Certified Core-Guided MaxSAT Solving

Andy Oertel

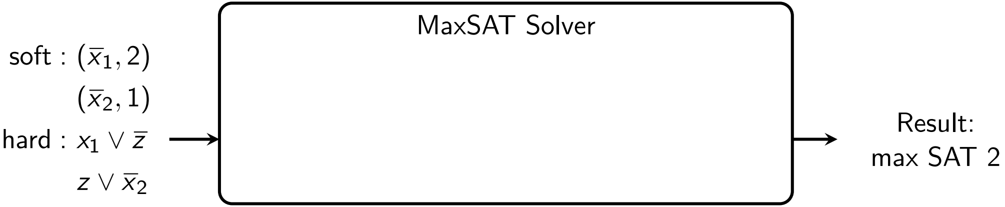
Lund University and
University of Copenhagen

Extended Reunion: Satisfiability 2023

April 24, 2023

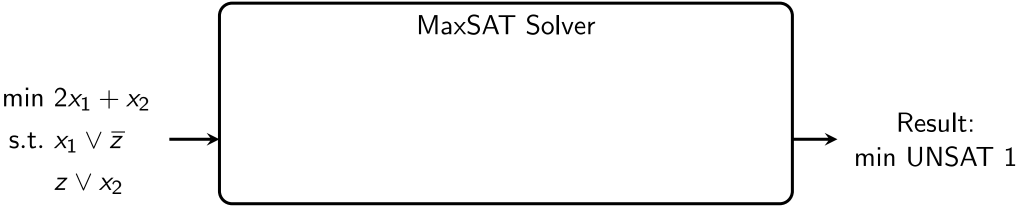
Joint work with Jeremias Berg, Bart Bogaerts, Jakob Nordström and Dieter Vandesande

Maximum Satisfiability (MaxSAT) Solving



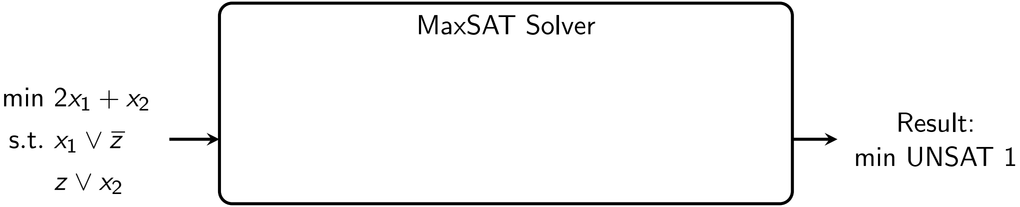
- Maximize sum of weight of satisfied soft clauses subject to satisfying hard clauses

Maximum Satisfiability (MaxSAT) Solving



- Maximize sum of weight of satisfied soft clauses subject to satisfying hard clauses
- Alternatively: Minimize linear objective subject to satisfying a CNF formula

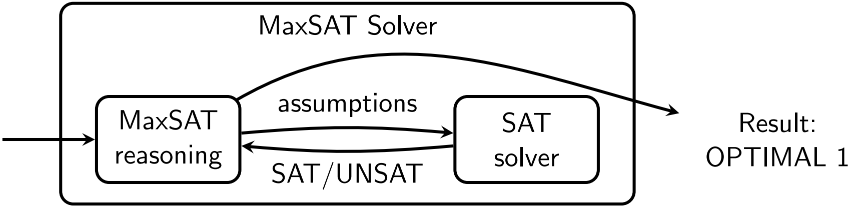
Maximum Satisfiability (MaxSAT) Solving



- Maximize sum of weight of satisfied soft clauses subject to satisfying hard clauses
- Alternatively: Minimize linear objective subject to satisfying a CNF formula
- Main approaches:
  - Solution-improving or linear SAT-UNSAT search [ES06, LP10, PRB18]
  - Implicit hitting set (IHS) search [DB13a, DB13b]
  - **Core-guided search** [FM06, NB14, ADR15, AG17]

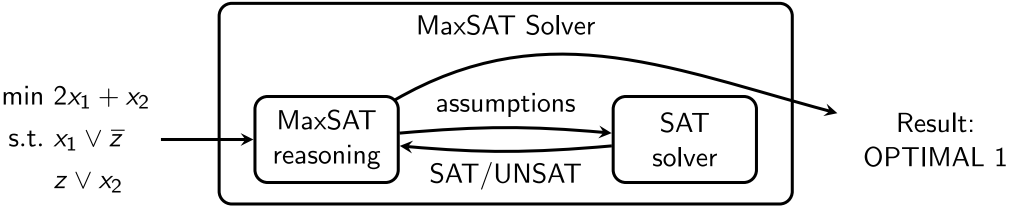
Core-Guided MaxSAT Solving

$$\min\ 2x_1 + x_2 \quad \text{s.t.}\ x_1 \vee \bar{z},\quad z \vee x_2$$



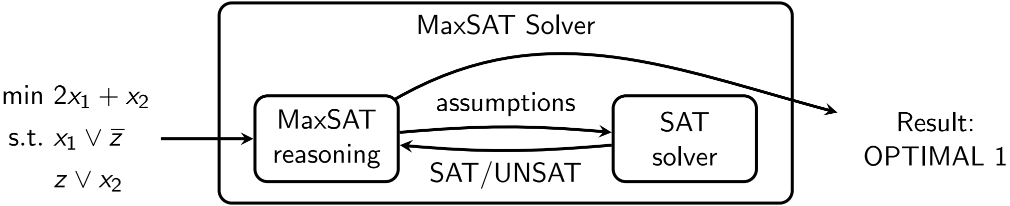
- Call SAT solver with optimistic assumptions (assignment to objective variables)

Core-Guided MaxSAT Solving



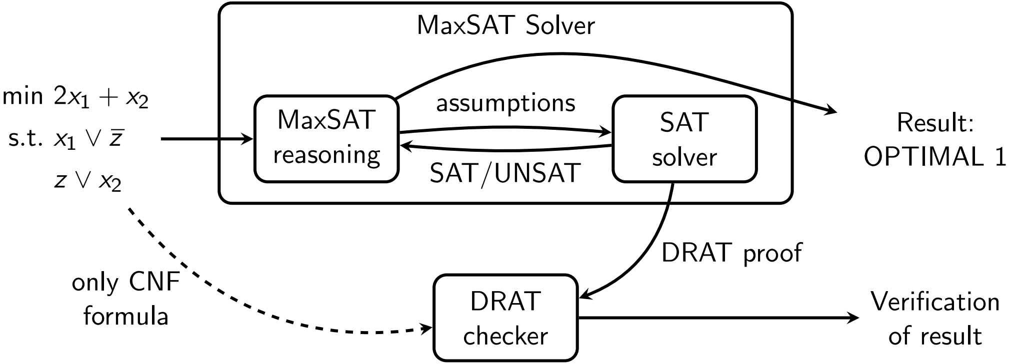
- Call SAT solver with optimistic assumptions (assignment to objective variables)
- If UNSAT, SAT solver returns core clause over assumptions explaining UNSAT
  - Reformulate objective, relax assumptions, increase lower bound
- If SAT, optimal solution found

Correctness of Core-Guided MaxSAT Solving



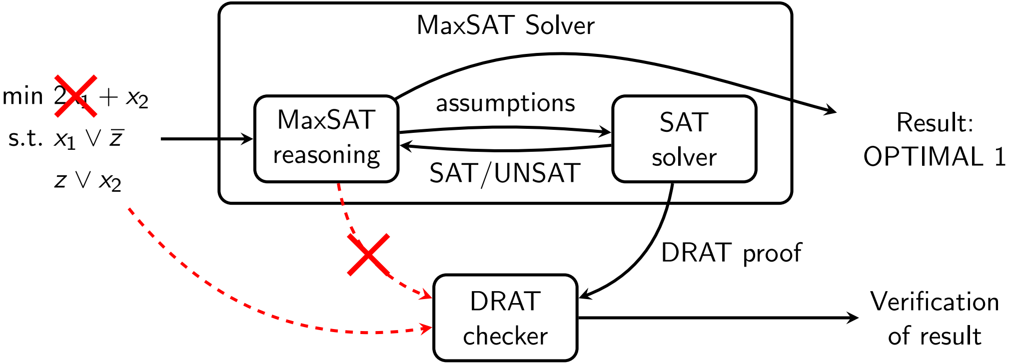
- Know cases of buggy MaxSAT solvers in MaxSAT evaluation

Correctness of Core-Guided MaxSAT Solving



- Know cases of buggy MaxSAT solvers in MaxSAT evaluation
- Correctness of SAT solver result can be certified [HHW13a, HHW13b, WHH14]

Correctness of Core-Guided MaxSAT Solving



- Know cases of buggy MaxSAT solvers in MaxSAT evaluation
- Correctness of SAT solver result can be certified [HHW13a, HHW13b, WHH14]
- **Core-guided MaxSAT reasoning not certified!**

Pseudo-Boolean (PB) Proof Logging

- **Multi-purpose** proof format
- Allows easy proof logging for
  - Reasoning with pseudo-Boolean constraints (by design)
  - SAT solving (including advanced techniques) [GN21, BGMN22]
  - Constraint programming [EGMN20, GMN22]
  - Subgraph problems [GMN20, GMM+20]
  - SAT-based pseudo-Boolean solving [GMNO22]
  - Unweighted linear SAT-UNSAT search MaxSAT [VDB22]

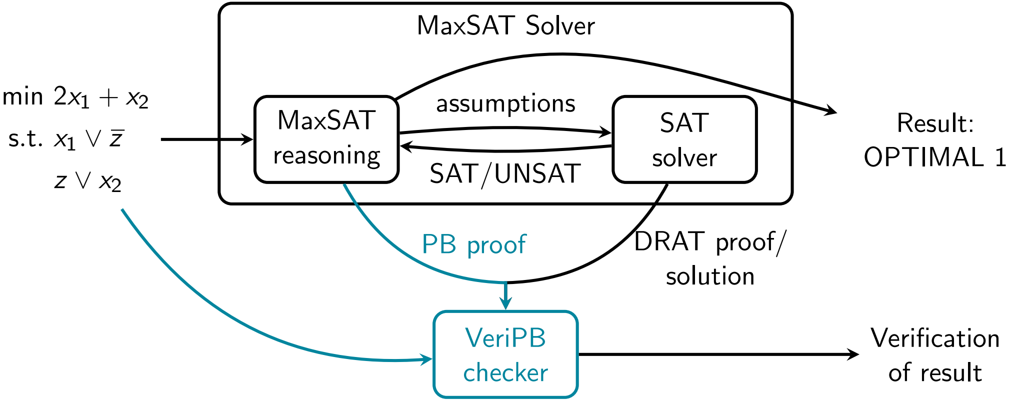
Pseudo-Boolean (PB) Proof Logging

- **Multi-purpose** proof format
- Allows easy proof logging for
  - Reasoning with pseudo-Boolean constraints (by design)
  - SAT solving (including advanced techniques) [GN21, BGMN22]
  - Constraint programming [EGMN20, GMN22]
  - Subgraph problems [GMN20, GMM$^+$20]
  - SAT-based pseudo-Boolean solving [GMNO22]
  - Unweighted linear SAT-UNSAT search MaxSAT [VDB22]

This work:

- Proof logging for state-of-the-art core-guided MaxSAT solving [IMM19, IBJ21]

Workflow



- Single PB proof that interleaves MaxSAT reasoning and SAT solver proof

Basic Notation

- Boolean variable x: with domain 0 (false) and 1 (true)
- Literal ℓ: x or negation $\overline{x} = 1 - x$
- Pseudo-Boolean (PB) constraint: integer linear inequality over literals

$$3x_1 + 2x_2 + 5\overline{x}_3 \geq 5$$

Basic Notation

- **Boolean variable x:** with domain 0 (false) and 1 (true)
- **Literal ℓ:** x or negation $\overline{x} = 1 - x$
- **Pseudo-Boolean (PB) constraint:** integer linear inequality over literals

$$3x_1 + 2x_2 + 5\overline{x}_3 \geq 5$$

- **Pseudo-Boolean equality constraint:** syntactic sugar for 2 inequalities

$$3x_1 + 2x_2 + 5\overline{x}_3 = 5 \longrightarrow \begin{array}{l} 3x_1 + 2x_2 + 5\overline{x}_3 \geq 5 \\ 3x_1 + 2x_2 + 5\overline{x}_3 \leq 5 \end{array}$$

Basic Notation

- Boolean variable x: with domain 0 (false) and 1 (true)
- Literal ℓ: x or negation $\overline{x} = 1 - x$
- Pseudo-Boolean (PB) constraint: integer linear inequality over literals

$$3x_1 + 2x_2 + 5\overline{x}_3 \geq 5$$

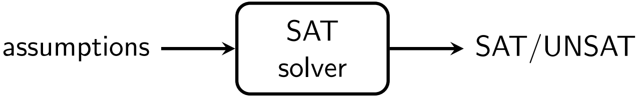
- Pseudo-Boolean equality constraint: syntactic sugar for 2 inequalities

$$3x_1 + 2x_2 + 5\overline{x}_3 = 5 \longrightarrow \begin{array}{l} 3x_1 + 2x_2 + 5\overline{x}_3 \geq 5 \\ 3x_1 + 2x_2 + 5\overline{x}_3 \leq 5 \end{array}$$

- Clause: disjunction of literals / at-least-one constraint

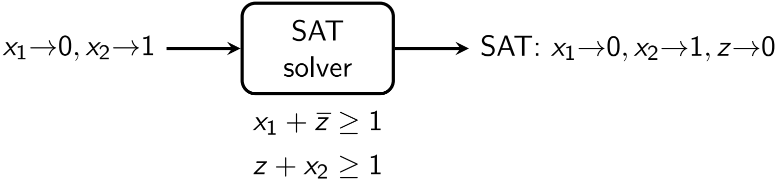
$$x_1 \vee \overline{x}_2 \vee \overline{x}_3 \iff x_1 + \overline{x}_2 + \overline{x}_3 \geq 1$$

Calling SAT Solver with Assumptions



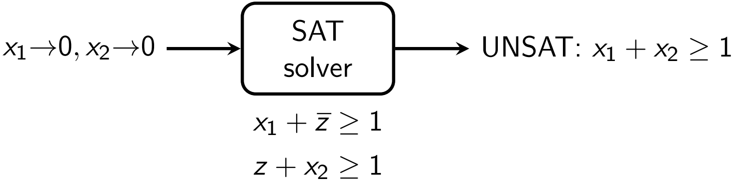
- Assumptions: partial assignment of the variables
- SAT solver tries to extend this partial assignment to a complete assignment

Calling SAT Solver with Assumptions



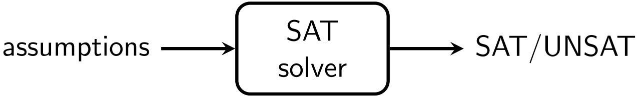
- Assumptions: partial assignment of the variables
- SAT solver tries to extend this partial assignment to a complete assignment
- SAT solver returns:
  - SAT: complete assignment

Calling SAT Solver with Assumptions



- Assumptions: partial assignment of the variables
- SAT solver tries to extend this partial assignment to a complete assignment
- SAT solver returns:
  - SAT: complete assignment
  - UNSAT: clause over assumption variables explaining why assumptions don't work

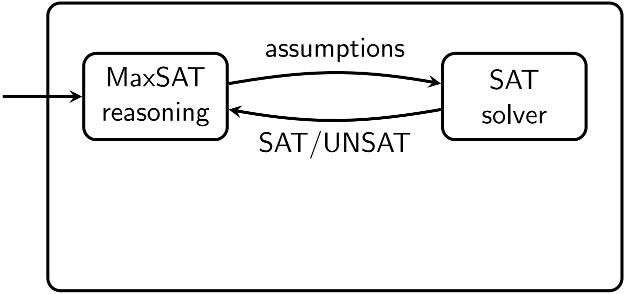
Calling SAT Solver with Assumptions



- Assumptions: partial assignment of the variables
- SAT solver tries to extend this partial assignment to a complete assignment
- SAT solver returns:
  - SAT: complete assignment
  - UNSAT: clause over assumption variables explaining why assumptions don't work
- Supported by modern SAT solvers

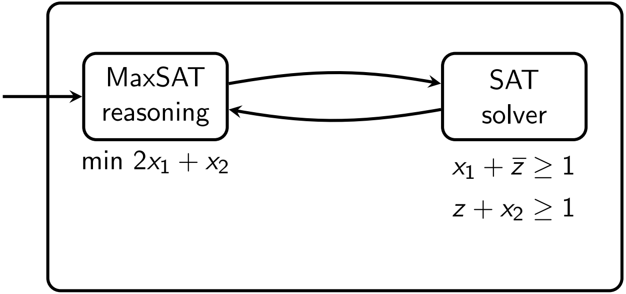
OLL-Style Core-Guided MaxSAT Solving [MDM14]

$$\min\ 2x_1 + x_2$$
$$\text{s.t. } x_1 + \bar{z} \geq 1$$
$$z + x_2 \geq 1$$

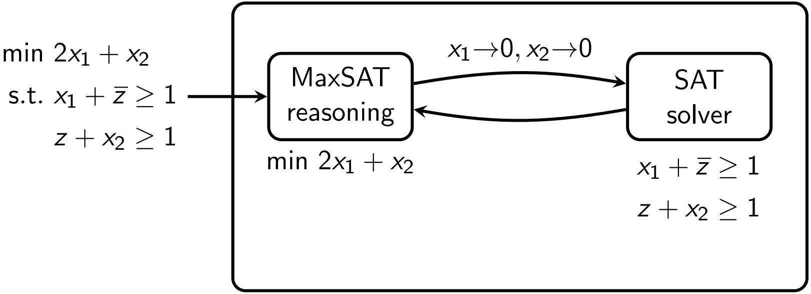


OLL-Style Core-Guided MaxSAT Solving [MDM14]

$$\min\ 2x_1 + x_2$$
$$\text{s.t.}\ x_1 + \bar{z} \geq 1$$
$$z + x_2 \geq 1$$

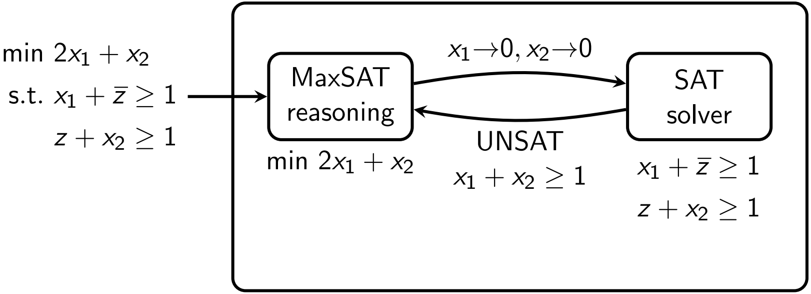


OLL-Style Core-Guided MaxSAT Solving [MDM14]



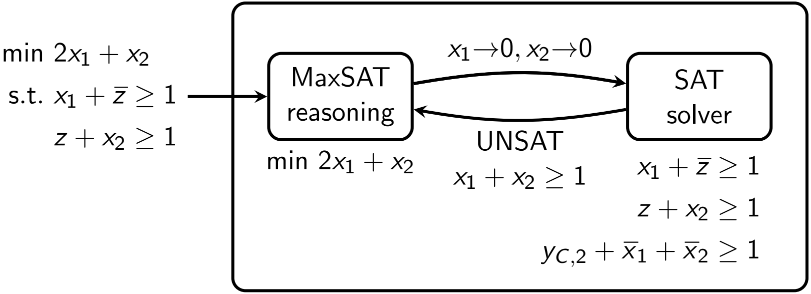
- Call with assumptions $x_1 \to 0, x_2 \to 0$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



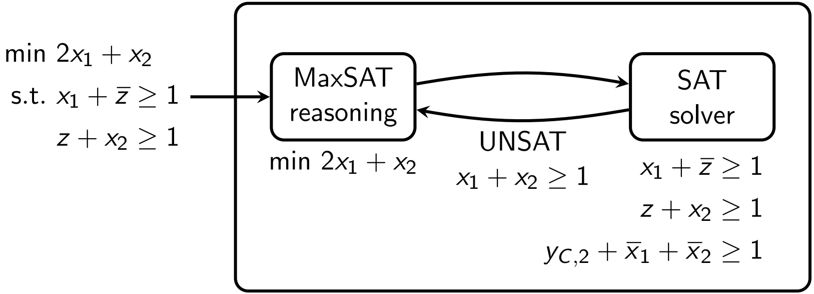
- Call with assumptions $x_1 \to 0, x_2 \to 0$
- SAT solver returns UNSAT with core $C : x_1 + x_2 \geq 1$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



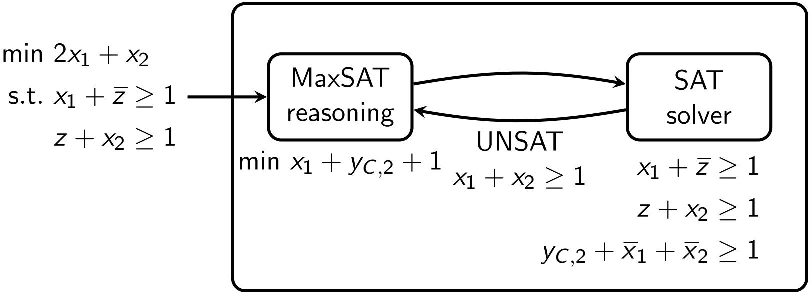
- Call with assumptions $x_1 \to 0, x_2 \to 0$
- SAT solver returns UNSAT with core $C : x_1 + x_2 \geq 1$
- Introduce counter variable $y_{C,2} \Leftrightarrow x_1 + x_2 \geq 2$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



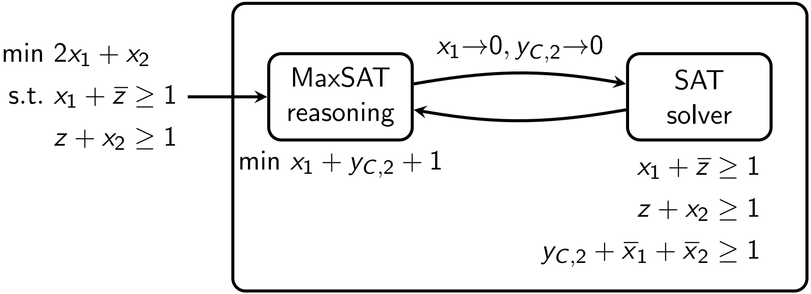
- Introduce counter variable $y_{C,2} \Leftrightarrow x_1 + x_2 \geq 2$
- Reformulate objective with $x_1 + x_2 = 1 + y_{C,2}$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



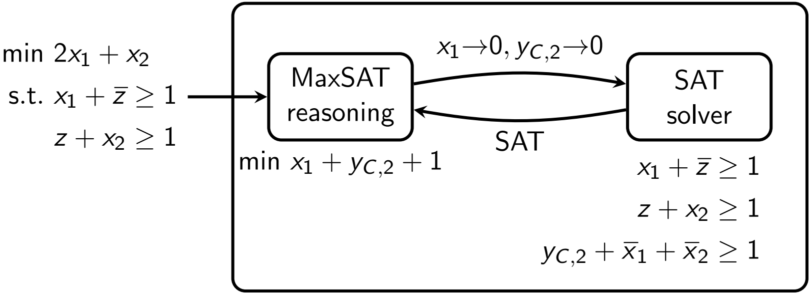
- Introduce counter variable $y_{C,2} \Leftrightarrow x_1 + x_2 \geq 2$
- Reformulate objective with $x_1 + x_2 = 1 + y_{C,2}$
- Substitute core literals in objective with counter variables

OLL-Style Core-Guided MaxSAT Solving [MDM14]



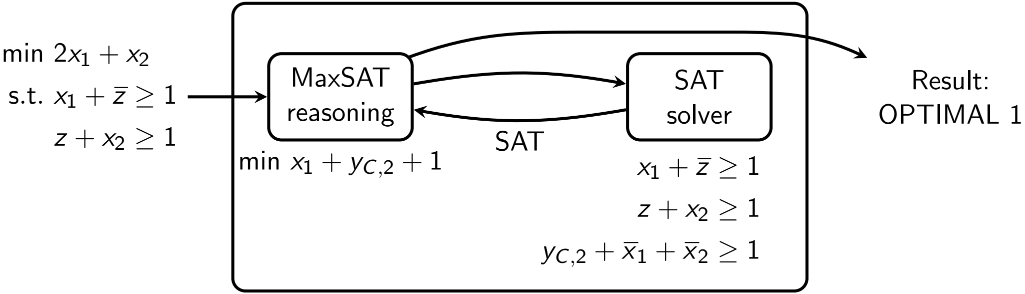
- Call with assumptions $x_1 \to 0, y_{C,2} \to 0$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



- Call with assumptions $x_1 \to 0, y_{C,2} \to 0$
- SAT solver returns SAT with solution $x_1 \to 0, x_2 \to 1, y_{C,2} \to 0, z \to 0$

OLL-Style Core-Guided MaxSAT Solving [MDM14]



- Call with assumptions $x_1 \to 0, y_{C,2} \to 0$
- SAT solver returns SAT with solution $x_1 \to 0, x_2 \to 1, y_{C,2} \to 0, z \to 0$
- Optimal solution found with value 1

Our Work: Correctness of Algorithm

- State-of-the-art solvers are way more complex [IMM19, IBJ21]
- Algorithm correct in theory [AKMS12, MDM14]
- But did we implement it correctly?

How can we show correctness of results?

Our Work: Correctness of Algorithm

- State-of-the-art solvers are way more complex [IMM19, IBJ21]
- Algorithm correct in theory [AKMS12, MDM14]
- But did we implement it correctly?

How can we show correctness of results?

- Proof logging!
- In this case: Formal derivation that the optimal value is optimal

Our Work: Correctness of Algorithm

- State-of-the-art solvers are way more complex [IMM19, IBJ21]
- Algorithm correct in theory [AKMS12, MDM14]
- But did we implement it correctly?

How can we show correctness of results?

- Proof logging!
- In this case: Formal derivation that the optimal value is optimal

This means

- Record optimal solution to proof $\implies$ upper bound on optimal value
- Derive lower bound on optimal value using PB reasoning

Our Work: Correctness of Algorithm

- State-of-the-art solvers are way more complex [IMM19, IBJ21]
- Algorithm correct in theory [AKMS12, MDM14]
- But did we implement it correctly?

How can we show correctness of results?

- Proof logging!
- In this case: Formal derivation that the optimal value is optimal

This means

- Record optimal solution to proof $\implies$ upper bound on optimal value
- Derive lower bound on optimal value using PB reasoning

Verification of core-guided MaxSAT solving!

Cutting Planes Proof System [CCT87]

Rules:

- Literal axiom

$$\frac{}{x \geq 0} \qquad \frac{}{\overline{x} \geq 0}$$

Cutting Planes Proof System [CCT87]

Rules:

- Literal axiom

$$\frac{}{x \geq 0} \qquad \frac{}{\overline{x} \geq 0}$$

- Addition

$$\text{Addition}\ \frac{x_1 + 2\overline{x}_2 + 2\overline{x}_3 \geq 3 \qquad \overline{x}_2 + 3x_3 \geq 3}{x_1 + 3\overline{x}_2 + x_3 \geq 4}$$

Cutting Planes Proof System [CCT87]

Rules:

- Literal axiom

$$\frac{}{x \geq 0} \qquad \frac{}{\overline{x} \geq 0}$$

- Addition

$$\text{Addition}\ \frac{x_1 + 2\overline{x}_2 + 2\overline{x}_3 \geq 3 \qquad \overline{x}_2 + 3x_3 \geq 3}{x_1 + 3\overline{x}_2 + x_3 \geq 4}$$

- Multiplication

$$\text{Multiply by 2}\ \frac{x_1 + 2\overline{x}_2 \geq 3}{2x_1 + 4\overline{x}_2 \geq 6}$$

Cutting Planes Proof System [CCT87]

Rules:

- Literal axiom

$$\frac{}{x \geq 0} \qquad \frac{}{\overline{x} \geq 0}$$

- Addition

$$\text{Addition}\ \frac{x_1 + 2\overline{x}_2 + 2\overline{x}_3 \geq 3 \qquad \overline{x}_2 + 3x_3 \geq 3}{x_1 + 3\overline{x}_2 + x_3 \geq 4}$$

- Multiplication

$$\text{Multiply by 2}\ \frac{x_1 + 2\overline{x}_2 \geq 3}{2x_1 + 4\overline{x}_2 \geq 6}$$

- Division (and rounding up)

$$\text{Divide by 2}\ \frac{2x_1 + 2\overline{x}_2 + 4x_3 \geq 5}{x_1 + \overline{x}_2 + 2x_3 \geq 2.5}$$

Cutting Planes Proof System [CCT87]

Rules:

- Literal axiom

$$\frac{}{x \geq 0} \qquad \frac{}{\overline{x} \geq 0}$$

- Addition

$$\text{Addition}\ \frac{x_1 + 2\overline{x}_2 + 2\overline{x}_3 \geq 3 \qquad \overline{x}_2 + 3x_3 \geq 3}{x_1 + 3\overline{x}_2 + x_3 \geq 4}$$

- Multiplication

$$\text{Multiply by 2}\ \frac{x_1 + 2\overline{x}_2 \geq 3}{2x_1 + 4\overline{x}_2 \geq 6}$$

- Division (and rounding up)

$$\text{Divide by 2}\ \frac{2x_1 + 2\overline{x}_2 + 4x_3 \geq 5}{x_1 + \overline{x}_2 + 2x_3 \geq 3}$$

Extended Cutting Planes: Reification

- Reification (special case of redundance rule in [GN21, BGMN22])

$$a \Leftrightarrow x_1 + \overline{x}_2 + 2x_3 \geq 2 \longrightarrow \begin{array}{ll} 2\overline{a} + x_1 + \overline{x}_2 + 2x_3 \geq 2 & (a \Rightarrow x_1 + \overline{x}_2 + 2x_3 \geq 2) \\ 3a + \overline{x}_1 + x_2 + 2\overline{x}_3 \geq 3 & (a \Leftarrow x_1 + \overline{x}_2 + 2x_3 \geq 2) \end{array}$$

- Variable a was not used in proof so far

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB
2. Establish lower bound LB

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB
2. Establish lower bound LB
   - Prove objective function reformulation (which contains LB)

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB
2. Establish lower bound LB
   - Prove objective function reformulation (which contains LB)
     - Show that core clauses are valid
     - Introduce clauses defining counter variables

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB
2. Establish lower bound LB
   - Prove objective function reformulation (which contains LB)
     - Show that core clauses are valid
     - Introduce clauses defining counter variables
3. Obtain proof of optimality if $LB = UB$

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

If SAT solver returns UNSAT:

- Core clause is implied by reverse unit propagation (RUP)

core: $x_1 + x_2 \geq 1$

formula:

$$x_1 + \bar{z} \geq 1$$
$$z + x_2 \geq 1$$

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

If SAT solver returns UNSAT:

- Core clause is implied by reverse unit propagation (RUP)

core: $x_1 + x_2 \geq 1$

formula:
$$x_1 + \bar{z} \geq 1$$
$$z + x_2 \geq 1$$

$$x_1 \to 0, \; x_2 \to 0$$

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

If SAT solver returns UNSAT:

- Core clause is implied by reverse unit propagation (RUP)

core: $x_1 + x_2 \geq 1$

formula:
$$x_1 + \bar{z} \geq 1$$
$$z + x_2 \geq 1$$

$$x_1 \to 0, \; x_2 \to 0, \; z \to 0$$

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

If SAT solver returns UNSAT:

- Core clause is implied by reverse unit propagation (RUP)

core: $x_1 + x_2 \geq 1$

formula: $x_1 + \bar{z} \geq 1$, $z + x_2 \geq 1$

$x_1 \to 0, \; x_2 \to 0, \; z \to 0$

Certifying SAT Solver

- SAT solvers already do DRAT proof logging — just rewrite to VeriPB syntax
- This proof is still sound when assumptions are used

If SAT solver returns UNSAT:

- Core clause is implied by reverse unit propagation (RUP)

core: $x_1 + x_2 \geq 1$ formula: $x_1 + \bar{z} \geq 1$, $z + x_2 \geq 1$

$$x_1 \to 0,\; x_2 \to 0,\; z \to 0$$

If SAT solver returns SAT:

- Solution is recorded in the proof

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB ✔
2. Establish lower bound LB
   - Prove objective function reformulation (which contains LB)
     - Show that core clauses are valid ✔
     - Introduce clauses defining counter variables
3. Obtain proof of optimality if $LB = UB$

Certifying Totalizers

- Counter variable $y_{C,j}$ for core C true iff $\sum_{\ell \in C} \ell \geq j$ ($y_{C,j} \Leftrightarrow \sum_{\ell \in C} \ell \geq j$)
- Convenient to define with pseudo-Boolean constraints using reification

Certifying Totalizers

- Counter variable $y_{C,j}$ for core C true iff $\sum_{\ell \in C} \ell \geq j$ ($y_{C,j} \Leftrightarrow \sum_{\ell \in C} \ell \geq j$)
- Convenient to define with pseudo-Boolean constraints using reification

$$C\colon\ x_1 + x_2 + x_3 \geq 1 \implies \begin{array}{rl} 2y_{C,2} + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 \geq 2 & (y_{C,2} \Leftarrow x_1 + x_2 + x_3 \geq 2) \\ 2\overline{y_{C,2}} + x_1 + x_2 + x_3 \geq 2 & (y_{C,2} \Rightarrow x_1 + x_2 + x_3 \geq 2) \\ y_{C,3} + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 \geq 1 & (y_{C,2} \Leftarrow x_1 + x_2 + x_3 \geq 3) \\ 3\bar{y}_{C,3} + x_1 + x_2 + x_3 \geq 3 & (y_{C,2} \Rightarrow x_1 + x_2 + x_3 \geq 3) \end{array}$$

Certifying Totalizers

- Counter variable $y_{C,j}$ for core C true iff $\sum_{\ell \in C} \ell \geq j$ ($y_{C,j} \Leftrightarrow \sum_{\ell \in C} \ell \geq j$)
- Convenient to define with pseudo-Boolean constraints using reification

$$C\colon\ x_1 + x_2 + x_3 \geq 1 \implies \begin{array}{rl} 2y_{C,2} + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 \geq 2 & (y_{C,2} \Leftarrow x_1 + x_2 + x_3 \geq 2) \\ 2\overline{y_{C,2}} + x_1 + x_2 + x_3 \geq 2 & (y_{C,2} \Rightarrow x_1 + x_2 + x_3 \geq 2) \\ y_{C,3} + \bar{x}_1 + \bar{x}_2 + \bar{x}_3 \geq 1 & (y_{C,2} \Leftarrow x_1 + x_2 + x_3 \geq 3) \\ 3\bar{y}_{C,3} + x_1 + x_2 + x_3 \geq 3 & (y_{C,2} \Rightarrow x_1 + x_2 + x_3 \geq 3) \end{array}$$

- Encode constraints from PB to CNF using totalizers
- Output of totalizers are counter variables
- Prove that clausal encoding follows from PB definition as in [GMNO22, VDB22]

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB ✔
2. Establish lower bound LB
   - Prove objective function reformulation (which contains LB)
     - Show that core clauses are valid ✔
     - Introduce clauses defining counter variables ✔
3. Obtain proof of optimality if $LB = UB$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min 2x_1 + x_2 \quad \overset{C : x_1 + x_2 \geq 1}{\Longrightarrow} \quad \min x_1 + y_{C,2} + 1$$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min 2x_1 + x_2 \quad \xrightarrow{C : x_1 + x_2 \geq 1} \quad \min x_1 + y_{C,2} + 1$$

$$2y_{C,1} + \bar{x}_1 + \bar{x}_2 \geq 2$$
$$y_{C,2} + \bar{x}_1 + \bar{x}_2 \geq 1$$

$$\bar{y}_{C,1} + x_1 + x_2 \geq 1$$
$$2\bar{y}_{C,2} + x_1 + x_2 \geq 2$$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min\ 2x_1 + x_2 \quad \overset{C\,:\,x_1 + x_2 \geq 1}{\Longrightarrow} \quad \min\ x_1 + y_{C,2} + 1$$

$$\bar{x}_1 + \bar{x}_2 \geq 0$$

$$y_{C,2} + \bar{x}_1 + \bar{x}_2 \geq 1$$

$$x_1 + x_2 \geq 1$$

$$2\bar{y}_{C,2} + x_1 + x_2 \geq 2$$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min 2x_1 + x_2 \quad \overset{C\,:\, x_1 + x_2 \geq 1}{\Longrightarrow} \quad \min x_1 + y_{C,2} + 1$$

$$\begin{aligned} \bar{x}_1 + \bar{x}_2 &\geq 0 \\ y_{C,2} + \bar{x}_1 + \bar{x}_2 &\geq 1 \end{aligned} \quad \Longrightarrow \quad x_1 + x_2 \leq 1 + y_{C,2}$$

$$\begin{aligned} x_1 + x_2 &\geq 1 \\ 2\bar{y}_{C,2} + x_1 + x_2 &\geq 2 \end{aligned} \quad \Longrightarrow \quad x_1 + x_2 \geq 1 + y_{C,2}$$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min 2x_1 + x_2 \quad \overset{C\,:\,x_1 + x_2 \geq 1}{\Longrightarrow} \quad \min x_1 + y_{C,2} + 1$$

$$\begin{array}{r} \bar{x}_1 + \bar{x}_2 \geq 0 \\ y_{C,2} + \bar{x}_1 + \bar{x}_2 \geq 1 \end{array} \Longrightarrow x_1 + x_2 \leq 1 + y_{C,2}$$

$$\Longrightarrow x_1 + x_2 = 1 + y_{C,2}$$

$$\begin{array}{r} x_1 + x_2 \geq 1 \\ 2\bar{y}_{C,2} + x_1 + x_2 \geq 2 \end{array} \Longrightarrow x_1 + x_2 \geq 1 + y_{C,2}$$

Objective Reformulation (Simplified Version)

- Maintain invariance $f_{orig} = f_{reform}$ as constraint
- For each core derive reformulation due to this core [GMNO22]

$$\min 2x_1 + x_2 \quad \overset{C : x_1 + x_2 \geq 1}{\Longrightarrow} \quad \min x_1 + y_{C,2} + 1$$

$$\left.\begin{array}{r} \bar{x}_1 + \bar{x}_2 \geq 0 \\ y_{C,2} + \bar{x}_1 + \bar{x}_2 \geq 1 \end{array}\right\} \Longrightarrow x_1 + x_2 \leq 1 + y_{C,2}$$

$$\left.\begin{array}{r} x_1 + x_2 \geq 1 \\ 2\bar{y}_{C,2} + x_1 + x_2 \geq 2 \end{array}\right\} \Longrightarrow x_1 + x_2 \geq 1 + y_{C,2}$$

$$\Longrightarrow x_1 + x_2 = 1 + y_{C,2}$$

- Add reformulation due to each core together to get $f_{orig} = f_{reform}$

Roadmap to Certifying Core-Guided MaxSAT

1. Record solution in proof to establish upper bound UB ✔
2. Establish lower bound LB ✔
   - Prove objective function reformulation (which contains LB) ✔
     - Show that core clauses are valid ✔
     - Introduce clauses defining counter variables ✔
3. Obtain proof of optimality if $LB = UB$

Proving Optimality

- **Idea:** Finding a better solution than the optimal solution is contradicting

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \implies 2x_1 + x_2 \leq 1 - 1$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\text{min } 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \implies 2x_1 + x_2 \leq 0$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \implies 2\bar{x}_1 + \bar{x}_2 \geq 3$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \implies 2\bar{x}_1 + \bar{x}_2 \geq 3$$

- Add objective-improving constraint and $f_{orig} = f_{reform}$

$$\text{Addition}\ \frac{2\bar{x}_1 + \bar{x}_2 \geq 3 \qquad x_1 + x_2 \geq 1 + y_{C,2}}{}$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min\ 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \implies 2\bar{x}_1 + \bar{x}_2 \geq 3$$

- Add objective-improving constraint and $f_{orig} = f_{reform}$

$$\text{Addition}\ \frac{2\bar{x}_1 + \bar{x}_2 \geq 3 \qquad x_1 + x_2 + \bar{y}_{C,2} \geq 2}{}$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\text{min } 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \quad \Longrightarrow \quad 2\overline{x}_1 + \overline{x}_2 \geq 3$$

- Add objective-improving constraint and $f_{orig} = f_{reform}$

$$\text{Addition}\ \frac{2\overline{x}_1 + \overline{x}_2 \geq 3 \qquad x_1 + x_2 + \overline{y}_{C,2} \geq 2}{\overline{x}_1 + \overline{y}_{C,2} \geq 3}$$

Proving Optimality

- Idea: Finding a better solution than the optimal solution is contradicting

- Optimal solution α recorded at least once in proof log
- Introduce objective-improving constraint that enforces $f_{orig} \leq f_{orig}(\alpha) - 1$

$$\min\ 2x_1 + x_2 \quad \text{and} \quad \text{solution: } x_1 \to 0,\ x_2 \to 1 \quad \Longrightarrow \quad 2\overline{x}_1 + \overline{x}_2 \geq 3$$

- Add objective-improving constraint and $f_{orig} = f_{reform}$

$$\text{Addition}\ \frac{2\overline{x}_1 + \overline{x}_2 \geq 3 \qquad x_1 + x_2 + \overline{y}_{C,2} \geq 2}{\overline{x}_1 + \overline{y}_{C,2} \geq 3}$$

- The result is contradicting

Certifying Advanced Techniques

- State-of-the-art core-guided MaxSAT solvers use additional techniques
  - Intrinsic at-most-one constraints [IMM19]
  - Hardening [ABGL12]
  - Lazy counter variables [MJML14]
- Proof logging also required for these techniques

Certifying Advanced Techniques

- State-of-the-art core-guided MaxSAT solvers use additional techniques
  - Intrinsic at-most-one constraints [IMM19]
  - Hardening [ABGL12]
  - Lazy counter variables [MJML14]
- Proof logging also required for these techniques

Very convenient with pseudo-Boolean reasoning!

Experiments

- Implemented certifying version of state-of-the-art solver CGSS[1] [IBJ21]
- Proof checked with proof checker VeriPB[2]
- Benchmarks from MaxSAT Evaluation 2022[3]
  - 607 unweighted instances and 594 weighted instances

[1] https://gitlab.com/MIAOresearch/software/certified-cgss
[2] https://gitlab.com/MIAOresearch/software/VeriPB
[3] https://maxsat-evaluations.github.io/2022/

Experiments

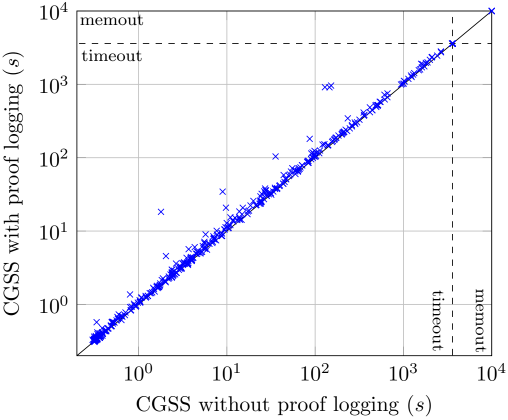
- Implemented certifying version of state-of-the-art solver CGSS[1] [IBJ21]
- Proof checked with proof checker VeriPB[2]
- Benchmarks from MaxSAT Evaluation 2022[3]
  - 607 unweighted instances and 594 weighted instances

First result:

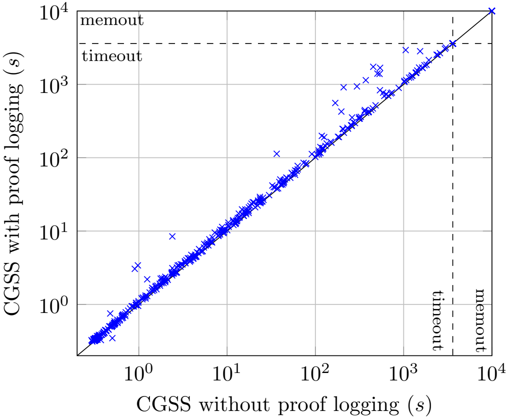
- Discovered bug in RC2 on which CGSS is based
  - Optimal solution correct for all instances in our set, but reasoning sometimes wrong
  - Can lead to solver claiming optimality for non-optimal solution

[1] `https://gitlab.com/MIAOresearch/software/certified-cgss`
[2] `https://gitlab.com/MIAOresearch/software/VeriPB`
[3] `https://maxsat-evaluations.github.io/2022/`

Overhead for Proof Logging in Seconds



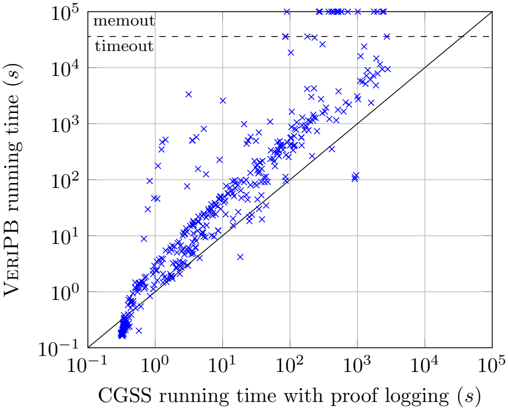
(a) Unweighted instances.



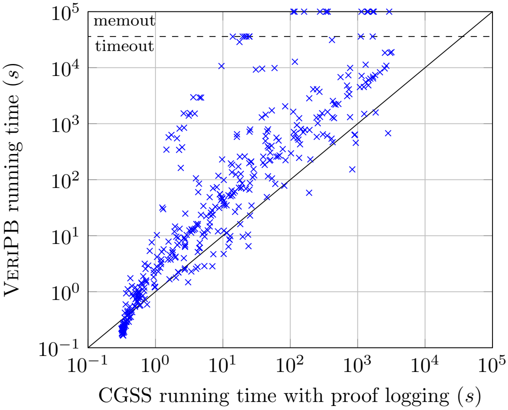
(b) Weighted instances.

- Low proof logging overhead (6.5% overhead in median)

Solving versus Checking in Seconds



(a) Unweighted instances.



(b) Weighted instances.

- Checking time could be improved (VeriPB not optimized for SAT solver proofs)

Future Work

Further proof logging:

- State-of-the-art linear SAT-UNSAT search solver (like Pacose)
- Implicit hitting set MaxSAT solver
  - Fundamental challenge: proof logging for MIP solver
- Pseudo-Boolean optimization

Future Work

Further proof logging:

- State-of-the-art linear SAT-UNSAT search solver (like Pacose)
- Implicit hitting set MaxSAT solver
  - Fundamental challenge: proof logging for MIP solver
- Pseudo-Boolean optimization

Improving performance:

- Optimize VeriPB for SAT solver proofs
- Backwards checking/trimming for verification (as in DRAT-trim [HHW13a])

Future Work

Further proof logging:

- State-of-the-art linear SAT-UNSAT search solver (like Pacose)
- Implicit hitting set MaxSAT solver
  - Fundamental challenge: proof logging for MIP solver
- Pseudo-Boolean optimization

Improving performance:

- Optimize VeriPB for SAT solver proofs
- Backwards checking/trimming for verification (as in DRAT-trim [HHW13a])

Sounds interesting? Join us! We are hiring!

Conclusion

- MaxSAT: successful optimization paradigm, but without proof logging
- DRAT not sufficient, but PB reasoning supports MaxSAT proof logging
- **This work:** Proof logging for state-of-the-art core-guided MaxSAT solving
- Hopefully step towards general adoption of proof logging for MaxSAT

Conclusion

- MaxSAT: successful optimization paradigm, but without proof logging
- DRAT not sufficient, but PB reasoning supports MaxSAT proof logging
- **This work:** Proof logging for state-of-the-art core-guided MaxSAT solving
- Hopefully step towards general adoption of proof logging for MaxSAT

Pseudo-Boolean reasoning provides unified proof logging method for:

- SAT solving (including advanced techniques) [GN21, BGMN22]
- Constraint programming [EGMN20, GMN22]
- Subgraph problems [GMN20, GMM$^+$20]
- SAT-based pseudo-Boolean solving [GMNO22]
- **This work:** Core-guided MaxSAT solving

Conclusion

- MaxSAT: successful optimization paradigm, but without proof logging
- DRAT not sufficient, but PB reasoning supports MaxSAT proof logging
- **This work:** Proof logging for state-of-the-art core-guided MaxSAT solving
- Hopefully step towards general adoption of proof logging for MaxSAT

Pseudo-Boolean reasoning provides unified proof logging method for:

- SAT solving (including advanced techniques) [GN21, BGMN22]
- Constraint programming [EGMN20, GMN22]
- Subgraph problems [GMN20, GMM+20]
- SAT-based pseudo-Boolean solving [GMNO22]
- **This work:** Core-guided MaxSAT solving
- **And soon(?):** All MaxSAT solving approaches

Conclusion

- MaxSAT: successful optimization paradigm, but without proof logging
- DRAT not sufficient, but PB reasoning supports MaxSAT proof logging
- **This work:** Proof logging for state-of-the-art core-guided MaxSAT solving
- Hopefully step towards general adoption of proof logging for MaxSAT

Pseudo-Boolean reasoning provides unified proof logging method for:

- SAT solving (including advanced techniques) [GN21, BGMN22]
- Constraint programming [EGMN20, GMN22]
- Subgraph problems [GMN20, GMM+20]
- SAT-based pseudo-Boolean solving [GMNO22]
- **This work:** Core-guided MaxSAT solving
- **And soon(?):** All MaxSAT solving approaches

Thank you for your attention!

References I

[ABGL12] Carlos Ansótegui, María Luisa Bonet, Joel Gabàs, and Jordi Levy.
Improving SAT-based weighted MaxSAT solvers.
In *Proceedings of the 18th International Conference on Principles and Practice of Constraint Programming (CP '12)*, volume 7514 of *Lecture Notes in Computer Science*, pages 86–101. Springer, October 2012.

[ADR15] Mario Alviano, Carmine Dodaro, and Francesco Ricca.
A maxsat algorithm using cardinality constraints of bounded size.
In Qiang Yang and Michael J. Wooldridge, editors, *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2677–2683. AAAI Press, 2015.

[AG17] Carlos Ansótegui and Joel Gabàs.
WPM3: an (in)complete algorithm for weighted partial maxsat.
Artif. Intell., 250:37–57, 2017.

References II

[AKMS12] Benjamin Andres, Benjamin Kaufmann, Oliver Matheis, and Torsten Schaub.
Unsatisfiability-based optimization in clasp.
In *Technical Communications of the 28th International Conference on Logic Programming (ICLP '12)*, volume 17 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 211–221, September 2012.

[BGMN22] Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström.
Certified symmetry and dominance breaking for combinatorial optimisation.
In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI '22)*, pages 3698–3707, February 2022.

[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.

[DB13a] Jessica Davies and Fahiem Bacchus.
Exploiting the power of MIP solvers in MAXSAT.
In *Proceedings of the 16th International Conference on Theory and Applications of Satisfiability Testing (SAT '13)*, volume 7962 of *Lecture Notes in Computer Science*, pages 166–181. Springer, July 2013.

References III

[DB13b] Jessica Davies and Fahiem Bacchus.
Postponing optimization to speed up MAXSAT solving.
In *CP*, volume 8124 of *Lecture Notes in Computer Science*, pages 247–262. Springer, 2013.

[EGMN20] Jan Elffers, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström.
Justifying all differences using pseudo-Boolean reasoning.
In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI '20)*, pages 1486–1494, February 2020.

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

[FM06] Zhaohui Fu and Sharad Malik.
On solving the partial MAX-SAT problem.
In *Proceedings of the 9th International Conference on Theory and Applications of Satisfiability Testing (SAT '06)*, volume 4121 of *Lecture Notes in Computer Science*, pages 252–265. Springer, August 2006.

References IV

[GMM+20] Stephan Gocht, Ross McBride, Ciaran McCreesh, Jakob Nordström, Patrick Prosser, and James Trimble.
Certifying solvers for clique and maximum common (connected) subgraph problems.
In *Proceedings of the 26th International Conference on Principles and Practice of Constraint Programming (CP '20)*, volume 12333 of *Lecture Notes in Computer Science*, pages 338–357. Springer, September 2020.

[GMN20] Stephan Gocht, Ciaran McCreesh, and Jakob Nordström.
Subgraph isomorphism meets cutting planes: Solving with certified solutions.
In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI '20)*, pages 1134–1140, July 2020.

[GMN22] Stephan Gocht, Ciaran McCreesh, and Jakob Nordström.
An auditable constraint programming solver.
In *Proceedings of the 28th International Conference on Principles and Practice of Constraint Programming (CP '22)*, volume 235 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:18, August 2022.

References V

[GMNO22] Stephan Gocht, Ruben Martins, Jakob Nordström, and Andy Oertel.
Certified CNF translations for pseudo-Boolean solving.
In *Proceedings of the 25th International Conference on Theory and Applications of Satisfiability Testing (SAT '22)*, volume 236 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:25, August 2022.

[GN21] Stephan Gocht and Jakob Nordström.
Certifying parity reasoning efficiently using pseudo-Boolean proofs.
In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI '21)*, pages 3768–3777, February 2021.

[HHW13a] Marijn J. H. Heule, Warren A. Hunt Jr., and Nathan Wetzler.
Trimming while checking clausal proofs.
In *Proceedings of the 13th International Conference on Formal Methods in Computer-Aided Design (FMCAD '13)*, pages 181–188, October 2013.

[HHW13b] Marijn J. H. Heule, Warren A. Hunt Jr., and Nathan Wetzler.
Verifying refutations with extended resolution.
In *Proceedings of the 24th International Conference on Automated Deduction (CADE-24)*, volume 7898 of *Lecture Notes in Computer Science*, pages 345–359. Springer, June 2013.

References VI

[IBJ21] Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo.
Refined core relaxation for core-guided maxsat solving.
In *27th International Conference on Principles and Practice of Constraint Programming (CP 2021)*, volume 210 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[IMM19] Alexey Ignatiev, António Morgado, and João P. Marques-Silva.
RC2: an efficient MaxSAT solver.
Journal on Satisfiability, Boolean Modeling and Computation, 11(1):53–64, September 2019.

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

[MDM14] António Morgado, Carmine Dodaro, and João P. Marques-Silva.
Core-guided MaxSAT with soft cardinality constraints.
In *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming (CP '14)*, volume 8656 of *Lecture Notes in Computer Science*, pages 564–573. Springer, September 2014.

References VII

[MJML14] Ruben Martins, Saurabh Joshi, Vasco M. Manquinho, and Inês Lynce.
Incremental cardinality constraints for MaxSAT.
In *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming (CP '14)*, volume 8656 of *Lecture Notes in Computer Science*, pages 531–548. Springer, September 2014.

[NB14] Nina Narodytska and Fahiem Bacchus.
Maximum satisfiability using core-guided maxsat resolution.
In *AAAI*, pages 2717–2723. AAAI Press, 2014.

[PRB18] Tobias Paxian, Sven Reimer, and Bernd Becker.
Dynamic polynomial watchdog encoding for solving weighted MaxSAT.
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 37–53. Springer, July 2018.

References VIII

[VDB22] Dieter Vandesande, Wolf De Wulf, and Bart Bogaerts.
QMaxSATpb: A certified MaxSAT solver.
In *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR '22)*, volume 13416 of *Lecture Notes in Computer Science*, pages 429–442. Springer, September 2022.

[WHH14] Nathan Wetzler, Marijn J. H. Heule, and Warren A. Hunt Jr.
DRAT-trim: Efficient checking and trimming using expressive clausal proofs.
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 422–429. Springer, July 2014.