Proof Logging for Maximum Satisfiability
the past, the present, the future

Jeremias Berg

HIIT, Department of Computer Science, University of Helsinki, Finland

April 20
Satisfiability: Theory, Practice, and Beyond
Simons Institute

Maximum Satisfiability

[Bacchus, Järvisalo, and Martins, 2021; Li and Manyà, 2021]

minimize: $cost \equiv \sum_i c_i \cdot b_i$

subject to: a set F of clauses

where: b_i boolean variables
$c_i > 0$ constants

- Competitive and thriving optimization paradigm
- New application domains and solver improvements annually.
  - Focus here on SAT-based MaxSAT solvers

Maximum Satisfiability

[Bacchus, Järvisalo, and Martins, 2021; Li and Manyà, 2021]

minimize: $cost \equiv \sum_i c_i \cdot b_i$

subject to: a set F of hard clauses

where: b_i boolean variables
$c_i > 0$ constants

Alternative	(and equivalent) definition:
minimize:	sum of weights of falsified soft clauses
where:	soft clauses: $\{(\neg b_i, c_i) \mid i = 1 \ldots\}$

- Competitive and thriving optimization paradigm
- New application domains and solver improvements annually.
  - Focus here on SAT-based MaxSAT solvers

Proof Logging

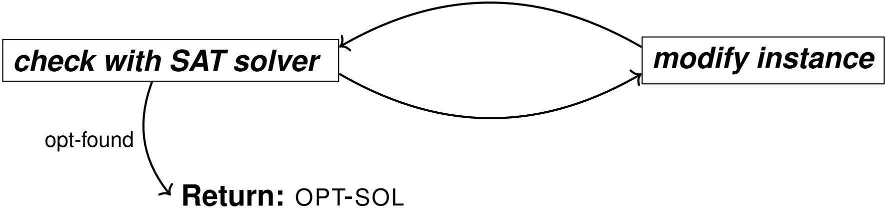
What?

Certificate of the optimal (minimum) cost of an instance.

Why?

- MaxSAT algorithms are complicated.
  - Implementations can be (and are) buggy.
- Increased trust enables new application domains.

(An oversimplification of) SAT-based MaxSAT



[Morgado, Dodaro, and Marques-Silva, 2014; Fu and Malik, 2006; Si, Zhang, Manquinho, Janota, Ignatiev, and Naik, 2016; Narodytska and Bacchus, 2014; Heras, Morgado, and Marques-Silva, 2011; Piotrów, 2020; Ignatiev, Morgado, and Marques-Silva, 2019; Ansótegui and Gabàs, 2017; Davies and Bacchus, 2011, 2013; Saikko, Berg, and Järvisalo, 2016; Paxian, Reimer, and Becker, 2018; Berre and Roussel, 2014; Koshimura, Zhang, Fujita, and Hasegawa, 2012]

Algorithms

Algorithms

Solution Improving

Upper-bounding search
with a SAT solver

Algorithms

Solution Improving

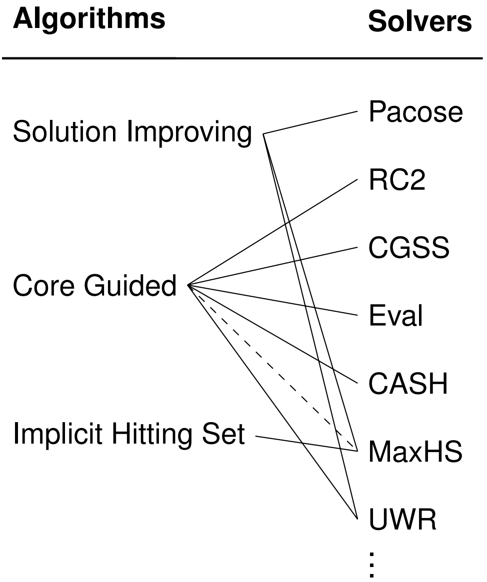
Upper-bounding search
with a SAT solver

Core Guided

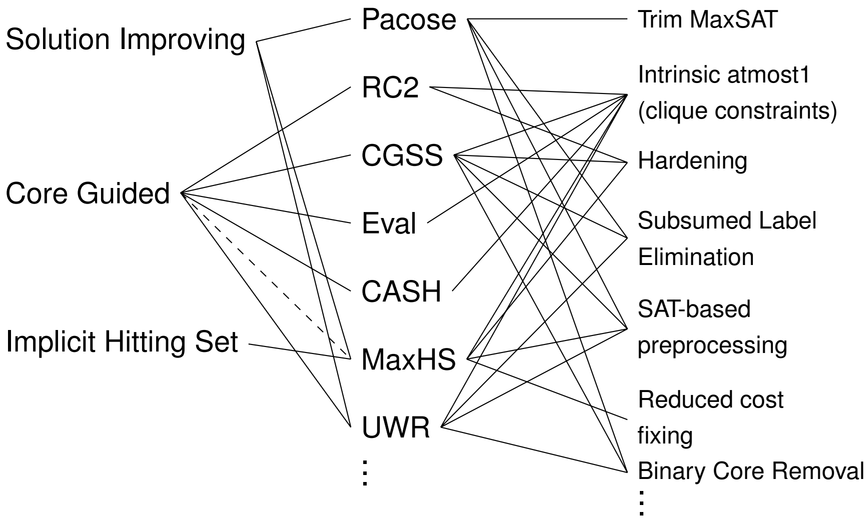
Lower-bounding search
with a SAT solver

Implicit Hitting Set

Lower-bounding search
with a SAT and MIP solver


Algorithms
Solvers
Solution Improving
Core Guided
Implicit Hitting Set
Pacose
RC2
CGSS
Eval
CASH
MaxHS
UWR
⋮

Algorithms **Solvers** **Heuristics**



Why not SAT proofs?

Intrinsic-at-most-ones / MuTexes / clique constraints

[Ignatiev, Morgado, and Marques-Silva, 2019]

$$
\begin{aligned}
cost &\equiv 2b_1 + 2b_2 + 4b_3 \\
F &= \{(b_1 \vee x), (\neg x \vee b_2), \\
&\qquad (b_2 \vee y), (\neg y \vee b_3), \\
&\qquad (b_1 \vee z), (\neg z \vee b_3)\}
\end{aligned}
$$

$$cost \geq 4$$

Why not SAT proofs?

Intrinsic-at-most-ones / MuTexes / clique constraints

[Ignatiev, Morgado, and Marques-Silva, 2019]

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$

$$F = \{(b_1 \vee x), (\neg x \vee b_2), (b_2 \vee y), (\neg y \vee b_3), (b_1 \vee z), (\neg z \vee b_3)\}$$

$$\longrightarrow F \models \bigwedge_{1 \leq i < j \leq 3} (b_i \vee b_j)$$

$$cost \geq 4$$

Why not SAT proofs?

Intrinsic-at-most-ones / MuTexes / clique constraints

[Ignatiev, Morgado, and Marques-Silva, 2019]

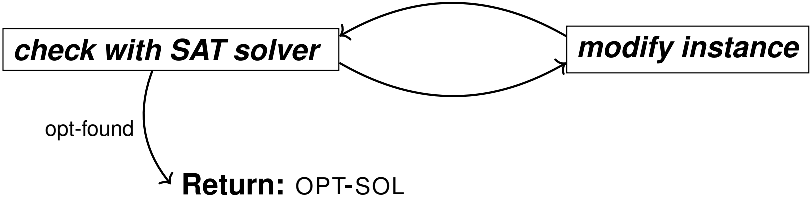
$$
\begin{aligned}
cost &\equiv 2b_1 + 2b_2 + 4b_3 \\
F &= \{(b_1 \vee x), (\neg x \vee b_2), \\
&\quad (b_2 \vee y), (\neg y \vee b_3), \\
&\quad (b_1 \vee z), (\neg z \vee b_3)\}
\end{aligned}
$$

$\longrightarrow$ $F \models \bigwedge_{1 \leq i < j \leq 3} (b_i \vee b_j)$

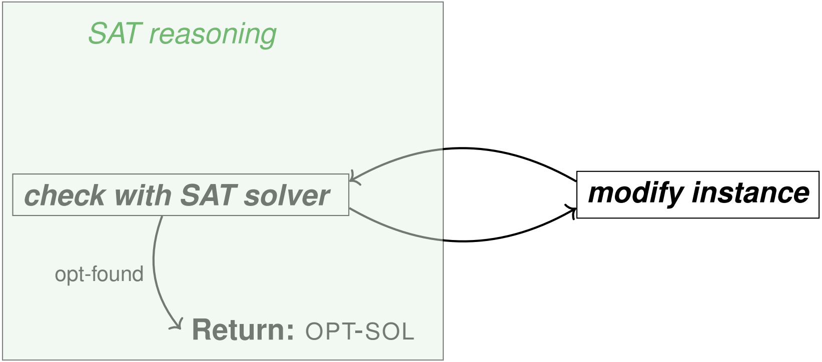
$\downarrow$

$cost \geq 4$???

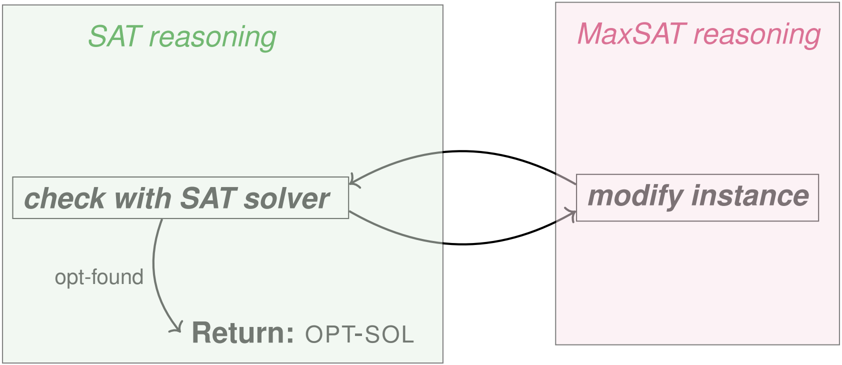
What is needed for proof logging SAT-based MaxSAT?



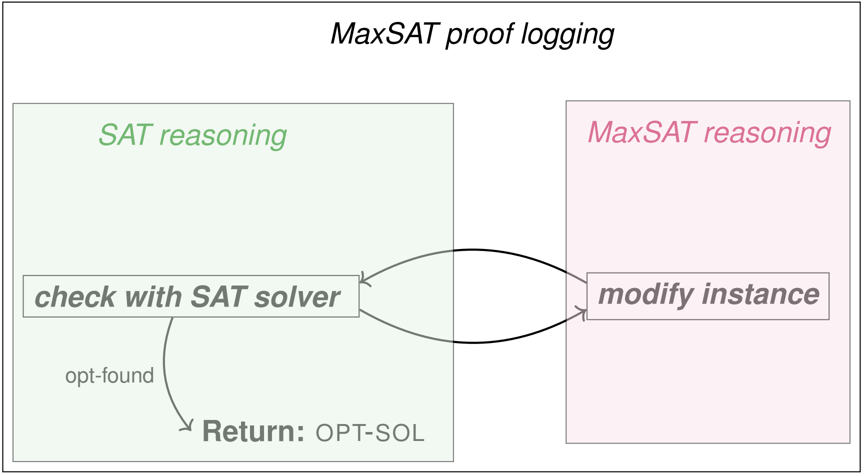
What is needed for proof logging SAT-based MaxSAT?



What is needed for proof logging SAT-based MaxSAT?



What is needed for proof logging SAT-based MaxSAT?



Take Home Messages

so far...

Modern SAT-based MaxSAT

- rich optimization paradigm
- many algorithms and heuristics
- extensive use of SAT solvers

Proof logging SAT-based MaxSAT requires

- proof logging SAT
- and reasoning with costs
- and supporting large diversity of techniques and algorithms

Take Home Messages

so far...

Modern SAT-based MaxSAT

- rich optimization paradigm
- many algorithms and heuristics
- extensive use of SAT solvers

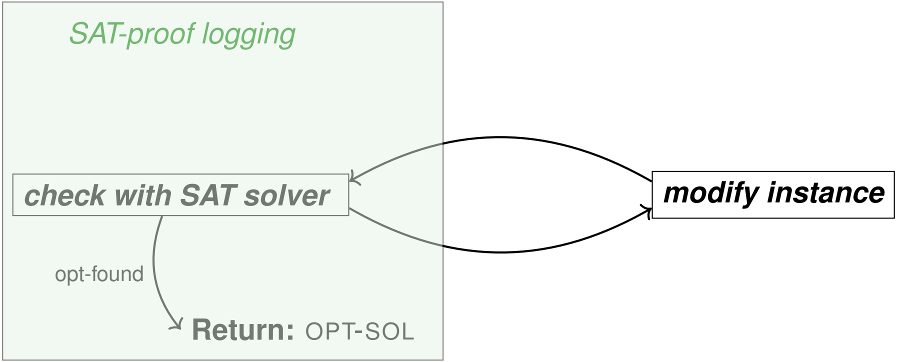
Proof logging SAT-based MaxSAT requires

- proof logging SAT
- and reasoning with costs
- and supporting large diversity of techniques and algorithms

Existing Approaches to Proof Logging MaxSAT

Checking SAT solvers

[Morgado and Marques-Silva, 2011]



- checking individual certificates returned by SAT solvers.
- frameworks for certifiable branch & bound search,
  - could be applied to modern B&B MaxSAT solvers [Abramé and Habet, 2016, Li, Xu, Coll, Manyà, Habet, and He, 2021]

Certifiable B&B

[Morgado and Marques-Silva, 2011][Larrosa, Nieuwenhuis, Oliveras, and Rodríguez-Carbonell, 2009, 2011]

Decide:	$\mathtt{I}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$	$\Rightarrow$	$\mathtt{I},\mathtt{l}^{d}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$
Unit Propagate:	$\mathtt{I}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$	$\Rightarrow$	$\mathtt{I}\mathtt{l}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$
Optimum:	$\mathtt{I}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$	$\Rightarrow$	*OptimumFound*
Backjump:	$\mathtt{I}\mathtt{l}^{d}\mathtt{I}'\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$	$\Rightarrow$	$\mathtt{I}\mathtt{l}\|\mathtt{S}\|\mathtt{k}\|\mathtt{A}$
Learn: ...			
Forget: ...			
Restart: ...			
Improve:			

- checking individual certificates returned by SAT solvers.
- frameworks for certifiable branch & bound search,
  - could be applied to modern B&B MaxSAT solvers [Abramé and Habet, 2016; Li, Xu, Coll, Manyà, Habet, and He, 2021]

Tableaux Calculus for MaxSAT

[Li, Manyà, and Soler, 2016; Li, Coll, Habet, Li, and Manyà, 2022; Li and Manyà, 2022]

Example:

$$F = \{(x \vee y, \top), (\neg x \vee b_1, \top), (\neg y \vee b_2, \top), (\neg b_1, 1), (\neg b_2, 1)\}$$

$\top \rightarrow$ hard clause

Tableaux Calculus for MaxSAT

[Li, Manyà, and Soler, 2016; Li, Coll, Habet, Li, and Manyà, 2022; Li and Manyà, 2022]

Example:

$$F = \{(x \vee y, \top), (\neg x \vee b_1, \top), (\neg y \vee b_2, \top), (\neg b_1, 1), (\neg b_2, 1)\}$$

$\top \rightarrow$hard clause

$$(x \vee y, \top)$$
$$|$$
$$(\neg x \vee b_1, \top)$$
$$|$$
$$(\neg y \vee b_2, \top)$$
$$|$$
$$(\neg b_1, 1)$$
$$|$$
$$(\neg b_2, 1)$$

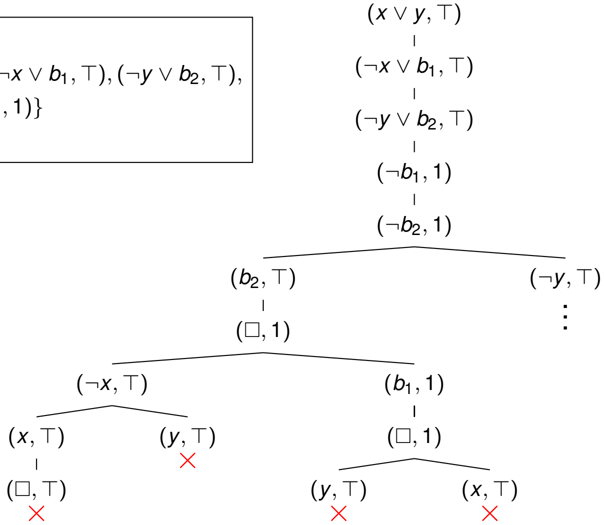
Tableaux Calculus for MaxSAT

[Li, Manyà, and Soler, 2016; Li, Coll, Habet, Li, and Manyà, 2022; Li and Manyà, 2022]

Example:

$F = \{(x \vee y, \top), (\neg x \vee b_1, \top), (\neg y \vee b_2, \top),$
$(\neg b_1, 1), (\neg b_2, 1)\}$

$\top$ $\rightarrow$hard clause



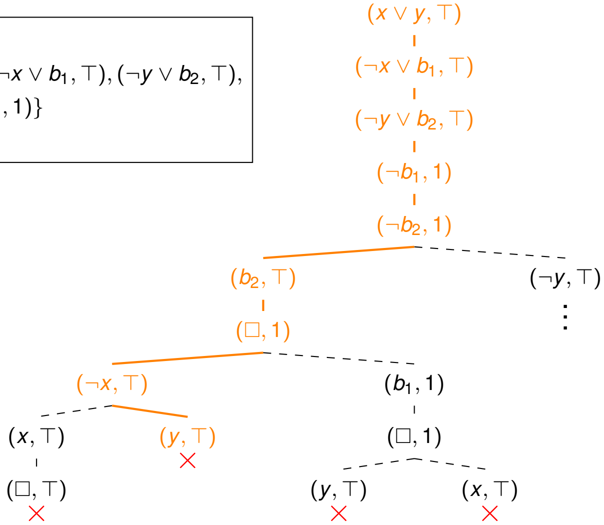
Tableaux Calculus for MaxSAT

[Li, Manyà, and Soler, 2016; Li, Coll, Habet, Li, and Manyà, 2022; Li and Manyà, 2022]

Example:

$$F = \{(x \vee y, \top), (\neg x \vee b_1, \top), (\neg y \vee b_2, \top), (\neg b_1, 1), (\neg b_2, 1)\}$$

$\top$ $\rightarrow$hard clause



Cost-aware clause redundancy notions

[Belov, Morgado, and Marques-Silva, 2013; Ihalainen, Berg, and Järvisalo, 2022; Berg and Järvisalo, 2019]

max-RAT

cost propagation redundancy

Cost-aware clause redundancy notions

[Belov, Morgado, and Marques-Silva, 2013; Ihalainen, Berg, and Järvisalo, 2022; Berg and Järvisalo, 2019]

max-RAT

- BCE, BVE, SE, SSR, BVA
- failed literals
- TrimMaxSAT
- and many more

*preserve **all** optimal solutions*

cost propagation redundancy

- group subsumed label elimination
- hardening

*preserve **one** optimal solution*

MaxSAT Resolution

[Larrosa and Heras, 2005; Bonet, Levy, and Manyà, 2007; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018; Py, Cherif, and Habet, 2020, 2022; Bjørner and Narodytska, 2015; Narodytska and Bacchus, 2014; Bonet, Buss, Ignatiev, Morgado, and Marques-Silva, 2021; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018]

$$\frac{(x \vee C, c_1) \quad (\neg x \vee D, c_2)}{\begin{array}{c} C \vee D, \mathrm{MIN}(c_1, c_2) \\ x \vee C \vee \neg D, c_1 - \mathrm{MIN}(c_1, c_2) \\ \neg x \vee D \vee \neg C, c_2 - \mathrm{MIN}(c_1, c_2) \\ x \vee C \vee \neg D, \mathrm{MIN}(c_1, c_2) \\ \neg x \vee \neg C \vee D, \mathrm{MIN}(c_1, c_2) \end{array}}$$

$$\frac{(x \vee C, \top) \quad (\neg x \vee D, c_2)}{\begin{array}{c} C \vee D, c_2 \\ x \vee C, \top \\ x \vee \neg C \vee D, c_2 \end{array}}$$

$$\frac{(x \vee C, \top) \quad (\neg x \vee D, \top)}{\begin{array}{c} C \vee D, \top \\ x \vee C, \top \\ \neg x \vee D, \top \end{array}}$$

$c_1, c_2 \in \mathbb{N}$,
$\top \rightarrow$ hard clause.

MaxSAT Resolution

[Larrosa and Heras, 2005; Bonet, Levy, and Manyà, 2007; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018; Py, Cherif, and Habet, 2020, 2022; Bjørner and Narodytska, 2015; Narodytska and Bacchus, 2014; Bonet, Buss, Ignatiev, Morgado, and Marques-Silva, 2021; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018]

$F_H \wedge F_S$

MaxSAT Resolution

[Larrosa and Heras, 2005; Bonet, Levy, and Manyà, 2007; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018; Py, Cherif, and Habet, 2020, 2022; Bjørner and Narodytska, 2015; Narodytska and Bacchus, 2014; Bonet, Buss, Ignatiev, Morgado, and Marques-Silva, 2021; Bonet, Buss, Ignatiev, Marques-Silva, and Morgado, 2018]

$$F_H \wedge F_S \xrightarrow{\textit{MaxRES steps}} F^* \wedge (\Box, \textsc{minimum-cost})$$

satisfiable

solutions are optimal of $F_h \wedge F_s$

Cutting Planes / VeriPB

[Gocht, 2022; Bogaerts, Gocht, McCreesh, and Nordström, 2022; Gocht and Nordström, 2021; Vandesande, De Wulf, and Bogaerts, 2022]

$$cost \equiv 2b_1 + 3b_2 + 4b_3$$

$$F = \{(b_1 \lor x), (\neg x \lor b_2), (b_2 \lor y), (\neg y \lor b_3), (b_1 \lor z), (\neg z \lor b_3)\}$$

$$\longrightarrow$$

$$cost \equiv 2b_1 + 3b_2 + 4b_3$$

$$F = \{b_1 + x \geq 1,\ b_2 + (1 - x) \geq 1,\ b_2 + y \geq 1,\ b_3 + (1 - y) \geq 1,\ b_1 + z \geq 1,\ b_3 + (1 - z) \geq 1\}$$

Cutting Planes for logging SAT-based MaxSAT

- Solution Improving (SAT/UNSAT)
  - specific solvers and PB encodings [Vandesande, De Wulf, and Bogaerts, 2022; Gocht, Martins, Nordström, and Oertel, 2022]
- Core-guided
  - specific solvers and algorithms
  - **Talk in SAT seminar on Monday**
- IHS
  - does not exist (yet...)
  - main challenge, hitting set computed by MIP

Cutting Planes for logging SAT-based MaxSAT

- Solution Improving (SAT/UNSAT)
  - specific solvers and PB encodings [Vandesande, De Wulf, and Bogaerts, 2022; Gocht, Martins, Nordström, and Oertel, 2022]
- Core-guided
  - specific solvers and algorithms
  - **Talk in SAT seminar on Monday**
- IHS
  - does not exist (yet...)
  - main challenge, hitting set computed by MIP

Cutting Planes for logging SAT-based MaxSAT

- Solution Improving (SAT/UNSAT)
  - specific solvers and PB encodings [Vandesande, De Wulf, and Bogaerts, 2022; Gocht, Martins, Nordström, and Oertel, 2022]
- Core-guided
  - specific solvers and algorithms
  - **Talk in SAT seminar on Monday**
- IHS
  - does not exist (yet...)
  - main challenge, hitting set computed by MIP

Future Directions & Challenges

- Effectively combining SAT and MaxSAT reasoning.
- Covering more techniques and solvers.
  - Mixed Integer Programming
- Making the techniques nice to use.
- Applications beyond complete solving and SAT-based solvers.
- Extending algorithmic ideas to other paradigms
  - ASP, PBO, CP ...

Future Directions & Challenges

- Effectively combining SAT and MaxSAT reasoning.
- Covering more techniques and solvers.
  - Mixed Integer Programming
- Making the techniques nice to use.
- Applications beyond complete solving and SAT-based solvers.
- Extending algorithmic ideas to other paradigms
  - ASP, PBO, CP . . .

Future Directions & Challenges

- Effectively combining SAT and MaxSAT reasoning.
- Covering more techniques and solvers.
  - Mixed Integer Programming
- Making the techniques nice to use.
- Applications beyond complete solving and SAT-based solvers.
- Extending algorithmic ideas to other paradigms
  - ASP, PBO, CP . . .

Summary

Maximum Satisfiability

- rich optimization paradigm
- large diversity of algorithms and heuristics

Summary

Maximum Satisfiability

- rich optimization paradigm
- large diversity of algorithms and heuristics

Proof logging SAT-based MaxSAT

- is **not** a straight-forward extension of proof logging SAT
- not **yet** as mature as SAT proof logging
  - recent promising developments
- many interesting future challenges

Bibliography I

André Abramé and Djamal Habet. Learning nobetter clauses in max-sat branch and bound solvers. In *ICTAI*, pages 452–459. IEEE Computer Society, 2016.

Carlos Ansótegui and Joel Gabàs. WPM3: An (in)complete algorithm for weighted partial MaxSAT. *Artificial Intelligence*, 250: 37–57, 2017. ISSN 0004-3702. doi: 10.1016/j.artint.2017.05.003.

Fahiem Bacchus, Matti Järvisalo, and Ruben Martins. Maximum satisfiability. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, Frontiers in Artificial Intelligence and Applications, chapter 24, pages 929–991. IOS Press, 2021.

A. Belov, A. Morgado, and J. Marques-Silva. SAT-based preprocessing for MaxSAT. In *Proc. LPAR-19*, volume 8312 of *Lecture Notes in Computer Science*, pages 96–111. Springer, 2013.

Jeremias Berg and Matti Järvisalo. Unifying reasoning and core-guided search for maximum satisfiability. In *JELIA*, volume 11468 of *Lecture Notes in Computer Science*, pages 287–303. Springer, 2019.

Daniel Le Berre and Stéphanie Roussel. Sat4j 2.3.2: on the fly solver configuration system description. *J. Satisf. Boolean Model. Comput.*, 8(3/4):197–202, 2014.

N. Bjørner and N. Narodytska. Maximum Satisfiability Using Cores and Correction Sets. In *Proc. IJCAI*, pages 246–252. AAAI Press, 2015.

Bart Bogaerts, Stephan Gocht, Ciaran McCreesh, and Jakob Nordström. Certified symmetry and dominance breaking for combinatorial optimisation. In *AAAI*, pages 3698–3707. AAAI Press, 2022.

Maria Luisa Bonet, Jordi Levy, and Felip Manyà. Resolution for Max-SAT. *Artificial Intelligence*, 171(8-9):606–618, 2007.

Maria Luisa Bonet, Sam Buss, Alexey Ignatiev, Joao Marques-Silva, and António Morgado. MaxSAT resolution with the dual rail encoding. In *AAAI*. AAAI Press, 2018.

Maria Luisa Bonet, Sam Buss, Alexey Ignatiev, António Morgado, and João Marques-Silva. Propositional proof systems based on maximum satisfiability. *Artif. Intell.*, 300:103552, 2021.

J. Davies and F. Bacchus. Solving MAXSAT by Solving a Sequence of Simpler SAT Instances. In *Proc. CP*, volume 6876 of *Lecture Notes in Computer Science*, pages 225–239. Springer, 2011.

J. Davies and F. Bacchus. Exploiting the power of MIP solvers in MaxSAT. In *Proc. SAT*, volume 7962 of *Lecture Notes in Computer Science*, pages 166–181. Springer, 2013.

Bibliography II

Z. Fu and S. Malik. On solving the partial MaxSAT problem. In *Proc. SAT*, volume 4121 of *Lecture Notes in Computer Science*, pages 252–265. Springer, 2006.

Stephan Gocht. *Certifying Correctness for Combinatorial Algorithms: by Using Pseudo-Boolean Reasoning*. PhD thesis, Lund University, Sweden, 2022.

Stephan Gocht and Jakob Nordström. Certifying parity reasoning efficiently using pseudo-boolean proofs. In *AAAI*, pages 3768–3777. AAAI Press, 2021.

Stephan Gocht, Ruben Martins, Jakob Nordström, and Andy Oertel. Certified CNF translations for pseudo-boolean solving. In *SAT*, volume 236 of *LIPIcs*, pages 16:1–16:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

F. Heras, A. Morgado, and J. Marques-Silva. Core-Guided Binary Search Algorithms for Maximum Satisfiability. In *Proc. AAAI*. AAAI Press, 2011.

Alexey Ignatiev, António Morgado, and João Marques-Silva. RC2: an efficient MaxSAT solver. *J. Satisf. Boolean Model. Comput.*, 11(1):53–64, 2019.

Hannes Ihalainen, Jeremias Berg, and Matti Järvisalo. Clause redundancy and preprocessing in maximum satisfiability. In *IJCAR*. Springer, 2022.

M. Koshimura, T. Zhang, H. Fujita, and R. Hasegawa. QMaxSAT: A Partial Max-SAT Solver. *Journal of Satisfiability, Boolean Modeling and Computation*, 8(1/2):95–100, 2012.

Javier Larrosa and Federico Heras. Resolution in Max-SAT and its relation to local consistency in weighted CSPs. In *Proc. IJCAI*, pages 193–198. Professional Book Center, 2005.

Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. Branch and bound for boolean optimization and the generation of optimality certificates. In *SAT*, volume 5584 of *Lecture Notes in Computer Science*, pages 453–466. Springer, 2009.

Javier Larrosa, Robert Nieuwenhuis, Albert Oliveras, and Enric Rodríguez-Carbonell. A framework for certified boolean branch-and-bound optimization. *J. Autom. Reason.*, 46(1):81–102, 2011.

Chu Min Li and Felip Manyà. Maxsat, hard and soft constraints. In *Handbook of Satisfiability*, volume 336 of *Frontiers in Artificial Intelligence and Applications*, pages 903–927. IOS Press, 2021.

Chu Min Li and Felip Manyà. Inference in maxsat and minsat. In *The Logic of Software. A Tasting Menu of Formal Methods*, volume 13360 of *Lecture Notes in Computer Science*, pages 350–369. Springer, 2022.

Bibliography III

Chu Min Li, Felip Manyà, and Joan Ramon Soler. A clause tableau calculus for MinSAT. In *Proc. CCIA*, volume 288 of *Frontiers in Artificial Intelligence and Applications*, pages 88–97. IOS Press, 2016.

Chu-Min Li, Zhenxing Xu, Jordi Coll, Felip Manyà, Djamal Habet, and Kun He. Combining clause learning and branch and bound for maxsat. In *CP*, volume 210 of *LIPIcs*, pages 38:1–38:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

Shoulin Li, Jordi Coll, Djamal Habet, Chu-Min Li, and Felip Manyà. A tableau calculus for maxsat based on resolution. In *CCIA*, volume 356 of *Frontiers in Artificial Intelligence and Applications*, pages 35–44. IOS Press, 2022.

A. Morgado, C. Dodaro, and J. Marques-Silva. Core-Guided MaxSAT with Soft Cardinality Constraints. In *Proc. CP*, volume 8656 of *Lecture Notes in Computer Science*, pages 564–573. Springer, 2014.

António Morgado and João Marques-Silva. On validating Boolean optimizers. In *IEEE 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011*, pages 924–926, 2011.

N. Narodytska and F. Bacchus. Maximum satisfiability using core-guided MaxSAT resolution. In *Proc. AAAI*, pages 2717–2723. AAAI Press, 2014.

Tobias Paxian, Sven Reimer, and Bernd Becker. Dynamic polynomial watchdog encoding for solving weighted maxsat. In *SAT*, volume 10929 of *Lecture Notes in Computer Science*, pages 37–53. Springer, 2018.

Marek Piotrów. Uwrmaxsat: Efficient solver for maxsat and pseudo-boolean problems. In *ICTAI*, pages 132–136. IEEE, 2020.

Matthieu Py, Mohamed Sami Cherif, and Djamal Habet. Towards bridging the gap between SAT and Max-SAT refutations. In *Proceedings of the 32nd IEEE International Conference on Tools with Artificial Intelligence (ICTAI '20)*, pages 137–144, November 2020.

Matthieu Py, Mohamed Sami Cherif, and Djamal Habet. Proofs and certificates for Max-SAT. *Journal of Artificial Intelligence Research*, 75:1373–1400, December 2022.

P. Saikko, J. Berg, and M. Järvisalo. LMHS: A SAT-IP hybrid MaxSAT solver. In *Proc. SAT*, volume 9710 of *Lecture Notes in Computer Science*, pages 539–546. Springer, 2016.

Xujie Si, Xin Zhang, Vasco M. Manquinho, Mikolás Janota, Alexey Ignatiev, and Mayur Naik. On incremental core-guided MaxSAT solving. In *Proc. CP*, volume 9892 of *Lecture Notes in Computer Science*, pages 473–482. Springer, 2016. doi: 10.1007/978-3-319-44953-1_30. URL https://doi.org/10.1007/978-3-319-44953-1_30.

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.

Running example

with VeriPB

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$

$$F = \{(b_1 \lor x), (\neg x \lor b_2), (b_2 \lor y), (\neg y \lor b_3), (b_1 \lor z), (\neg z \lor b_3)\}$$

Running example

with VeriPB

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$

$$\begin{aligned} F = \{ & b_1 + x \geq 1, (1 - x) + b_2 \geq 1, \\ & b_2 + y \geq 1, (1 - y) + b_3 \geq 1, \\ & b_1 + z \geq 1, (1 - z) + b_3 \geq 1 \} \end{aligned}$$

Running example

with VeriPB

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$
$$F = \{b_1 + x \geq 1, (1 - x) + b_2 \geq 1, b_2 + y \geq 1, (1 - y) + b_3 \geq 1, b_1 + z \geq 1, (1 - z) + b_3 \geq 1\}$$

$\longrightarrow$

$$cost \equiv 2b_1 + 3b_2 + 4b_3$$
$$F \wedge \bigwedge_{1 \leq i < j \leq 3} b_i + b_j \geq 1$$

Running example

with VeriPB

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$

$$F = \{b_1 + x \geq 1, (1-x) + b_2 \geq 1, b_2 + y \geq 1, (1-y) + b_3 \geq 1, b_1 + z \geq 1, (1-z) + b_3 \geq 1\}$$

→

$$cost \equiv 2b_1 + 3b_2 + 4b_3$$

$$F \wedge \bigwedge_{1 \leq i < j \leq 3} b_i + b_j \geq 1$$

↓

$$cost \equiv 2b_1 + 2b_2 + 4b_3,$$

$$F \wedge \bigwedge_{1 \leq i < j \leq 3} b_i + b_j \geq 1 \wedge$$

$$* b_1 + b_2 + b_3 \geq 2 \wedge$$

$$2b_1 + 2b_2 + 2b_3 \geq 4 \wedge$$

$$cost \geq 2b_1 + 2b_2 + 2b_3$$

Running example

with VeriPB

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$
$$F = \{b_1 + x \geq 1, (1 - x) + b_2 \geq 1, b_2 + y \geq 1, (1 - y) + b_3 \geq 1, b_1 + z \geq 1, (1 - z) + b_3 \geq 1\}$$

$\longrightarrow$

$$cost \equiv 2b_1 + 3b_2 + 4b_3$$
$$F \wedge \bigwedge_{1 \leq i < j \leq 3} b_i + b_j \geq 1$$

$\downarrow$

$$cost \equiv 2b_1 + 2b_2 + 4b_3,$$
$$F \wedge \bigwedge_{1 \leq i < j \leq 3} b_i + b_j \geq 1 \wedge$$
$$* b_1 + b_2 + b_3 \geq 2 \wedge$$
$$2b_1 + 2b_2 + 2b_3 \geq 4 \wedge$$
$$cost \geq 2b_1 + 2b_2 + 2b_3$$

$\longleftarrow$

$$cost \geq 4$$

Running example

with MaxSAT resolution

$$cost \equiv 2b_1 + 2b_2 + 4b_3$$

$$F = \{(b_1 \vee x), (\neg x \vee b_2), (b_2 \vee y), (\neg y \vee b_3), (b_1 \vee z), (\neg z \vee b_3)\}$$

Running example

with MaxSAT resolution

$$
\begin{aligned}
F = \{&(b_1 \vee x, \top), (\neg x \vee b_2, \top),\\
&(b_2 \vee y, \top), (\neg y \vee b_3, \top),\\
&(b_1 \vee z, \top), (\neg z \vee b_3, \top)\\
&(\neg b_1, 2), (\neg b_2, 2), (\neg b_3, 4)\}
\end{aligned}
$$

Running example

with MaxSAT resolution

$$F = \{(b_1 \vee x, \top), (\neg x \vee b_2, \top), (b_2 \vee y, \top), (\neg y \vee b_3, \top), (b_1 \vee z, \top), (\neg z \vee b_3, \top) (\neg b_1, 2), (\neg b_2, 2), (\neg b_3, 4)\}$$

$$\longrightarrow F \wedge \bigwedge_{1 \leq i < j \leq 3} ((b_i \vee b_j), \top)$$

$$\downarrow *$$

$$F^* = \{(b_1 \vee x, \top), (\neg x \vee b_2, \top), (b_2 \vee y, \top), (\neg y \vee b_3, \top), (b_1 \vee z, \top), (\neg z \vee b_3, \top), (b_1 \vee b_2, \top), (b_1 \vee b_3, \top), (b_2 \vee b_3, \top), (\neg b_1 \vee \neg b_2 \vee \neg b_3, 1), (b_2 \vee b_3, 2), (b_1 \vee b_2 \vee b_3, 2), (\square, 4)\}$$

Running example

with MaxSAT resolution

$$F = \{(b_1 \vee x, \top), (\neg x \vee b_2, \top), (b_2 \vee y, \top), (\neg y \vee b_3, \top), (b_1 \vee z, \top), (\neg z \vee b_3, \top) (\neg b_1, 2), (\neg b_2, 2), (\neg b_3, 4)\}$$

$$\longrightarrow F \wedge \bigwedge_{1 \leq i < j \leq 3} ((b_i \vee b_j), \top)$$

$$\downarrow *$$

$$F^* = \{(b_1 \vee x, \top), (\neg x \vee b_2, \top), (b_2 \vee y, \top), (\neg y \vee b_3, \top), (b_1 \vee z, \top), (\neg z \vee b_3, \top), (b_1 \vee b_2, \top), (b_1 \vee b_3, \top), (b_2 \vee b_3, \top), (\neg b_1 \vee \neg b_2 \vee \neg b_3, 1), (b_2 \vee b_3, 2), (b_1 \vee b_2 \vee b_3, 2), (\square, 4)\}$$

$$cost \geq 4 \longleftarrow$$