Preprocessing in Incomplete MaxSAT Solving

Marcus Leivo Jeremias Berg Matti Järvisalo

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

ECAI 2020
Online

Contributions

- We analyse the effects of preprocessing on (incomplete) MaxSAT solving.
  - Using preprocessing results in solvers misinterpreting cost of *non-optimal solutions* both in theory and practice.
- Propose *locally minimal solutions*
  - A sufficient criterion for avoiding misinterpretation of cost.
- Develop stochastic local search (SLS) over locally minimal solutions
  - Searching over locally minimal solutions improves performance in practice.

Contributions

- We analyse the effects of preprocessing on (incomplete) MaxSAT solving.
  - Using preprocessing results in solvers misinterpreting cost of *non-optimal solutions* both in theory and practice.
- Propose *locally minimal solutions*
  - A sufficient criterion for avoiding misinterpretation of cost.
- Develop stochastic local search (SLS) over locally minimal solutions
  - Searching over locally minimal solutions improves performance in practice.

Contributions

- We analyse the effects of preprocessing on (incomplete) MaxSAT solving.
  - Using preprocessing results in solvers misinterpreting cost of *non-optimal solutions* both in theory and practice.
- Propose *locally minimal solutions*
  - A sufficient criterion for avoiding misinterpretation of cost.
- Develop stochastic local search (SLS) over locally minimal solutions
  - Searching over locally minimal solutions improves performance in practice.

Outline

- Motivation - MaxSAT
- Preprocessing and Cost Preservation
- Locally Minimal Solutions
- Locally Minimal Solutions in SLS MaxSAT solving
- Empirical Evaluation

Motivation: MaxSAT

- A declarative optimisation paradigm based on propositional logic.
- Various types of efficient MaxSAT solvers have been developed.
- Enable efficiently solving various NP-hard optimization problems via *encode-and-solve*.
  - Applications in:
    - AI & ML
    - Data Analysis
    - Industry
    - and many other domains...

Motivation: MaxSAT

- A declarative optimisation paradigm based on propositional logic.
- Various types of efficient MaxSAT solvers have been developed.
- Enable efficiently solving various NP-hard optimization problems via *encode-and-solve*.

- Applications in:
  - AI & ML
  - Data Analysis
  - Industry
  - and many other domains...

Motivation: MaxSAT

- A declarative optimisation paradigm based on propositional logic.
- Various types of efficient MaxSAT solvers have been developed.
- Enable efficiently solving various NP-hard optimization problems via *encode-and-solve*.

- Applications in:
  - AI & ML
  - Data Analysis
  - Industry
  - and many other domains...

Ghosh and Meel [2019]
Chen et al. [2010]
Zhang and Bacchus [2012]
Berg and Järvisalo [2017]
Demirovic et al. [2019]
Hosokawa et al. [2019]
Bacchus et al. [2019]
...

Motivation: MaxSAT

- Solving a MaxSAT instance $\mathcal{F}$
  - Find truth assignment that satisfies all hard clauses in $\mathcal{F}_H$ and maximises the number of satisfied soft clauses in $\mathcal{F}_S$.
  - Can have weights on soft clauses.

$$\mathcal{F}_H = \{(b_1 \vee b_2), (b_2 \vee b_3)\}$$

$$\mathcal{F}_S = \{(\neg b_1), (\neg b_2), (\neg b_3)\}$$

Motivation: MaxSAT

- Solving a MaxSAT instance $\mathcal{F}$
  - Find truth assignment that satisfies all hard clauses in $\mathcal{F}_H$ and maximises the number of satisfied soft clauses in $\mathcal{F}_S$.
  - Can have weights on soft clauses.

$$\mathcal{F}_H = \{(b_1 \vee b_2), (b_2 \vee b_3)\}$$

$$\mathcal{F}_S = \{(\neg b_1), (\neg b_2), (\neg b_3)\}$$

$$\tau = \{\neg b_1, b_2, \neg b_3\}$$

$$cost(\tau) = \mathbf{1}$$

Two Topical Research Directions

Incomplete Solving

Aim at quickly computing a solution that is as good as possible (not necessarily an optimal one)

Two Topical Research Directions

Incomplete Solving

Aim at quickly computing a solution that is as good as possible (not necessarily an optimal one)

Preprocessing

The application of simplification and deduction rules to an instance prior to search.

Two Topical Research Directions

Incomplete Solving

Aim at quickly computing a solution that is as good as possible (not necessarily an optimal one)

Preprocessing

The application of simplification and deduction rules to an instance prior to search.

This work:

Focus on understanding how costs of non-optimal solutions are influenced by preprocessing

Two Topical Research Directions

This work:
Focus on understanding how costs of non-optimal solutions are influenced by preprocessing

Note

Non-optimal solutions encountered both in complete, and incomplete solving

MaxSAT Preprocessing

Preprocessing an instance $\mathcal{F}$:

$$\mathcal{F}_H = \{ \quad (x \vee \neg y), (\neg x \vee y) \quad \}$$
$$\mathcal{F}_S = \{ \quad (\neg x) \quad \}$$

MaxSAT Preprocessing

Preprocessing an instance $\mathcal{F}$:

1. Apply preprocessing rules to form $\textsc{Pre}(\mathcal{F})$
2. Solve $\textsc{Pre}(\mathcal{F})$
3. Reconstruct a solution to $\mathcal{F}$

$\mathcal{F}_H = \{ \ (x \vee \neg y), (\neg x \vee y) \ \}$
$\mathcal{F}_S = \{ \ (\neg x) \ \}$

$\textsc{Pre}(\mathcal{F}_H) = \{ \ (x \vee \neg y) \ \}$
$\textsc{Pre}(\mathcal{F}_S) = \{ \ (\neg x) \ \}$

MaxSAT Preprocessing

Preprocessing an instance $\mathcal{F}$:

1. Apply preprocessing rules to form $\textsc{Pre}(\mathcal{F})$
2. Solve $\textsc{Pre}(\mathcal{F})$
3. Reconstruct a solution to $\mathcal{F}$

$\mathcal{F}_H = \{ \ (x \vee \neg y), (\neg x \vee y) \ \}$
$\mathcal{F}_S = \{ \ (\neg x) \ \}$

$\textsc{Pre}(\mathcal{F}_H) = \{ \ (x \vee \neg y) \ \}$
$\textsc{Pre}(\mathcal{F}_S) = \{ \ (\neg x) \ \}$

$\tau = \{\neg x, \neg y\}$

MaxSAT Preprocessing

Preprocessing an instance $\mathcal{F}$:

1. Apply preprocessing rules to form $\textsc{Pre}(\mathcal{F})$
2. Solve $\textsc{Pre}(\mathcal{F})$
3. Reconstruct a solution to $\mathcal{F}$

$\mathcal{F}_H = \{ \quad (x \vee \neg y), (\neg x \vee y) \quad \}$
$\mathcal{F}_S = \{ \quad (\neg x) \quad \}$

$\textsc{Pre}(\mathcal{F}_H) = \{ \quad (x \vee \neg y) \quad \}$
$\textsc{Pre}(\mathcal{F}_S) = \{ \quad (\neg x) \quad \}$

$\textsc{Rec}(\tau) = \{\neg x, \neg y\}$

$\tau = \{\neg x, \neg y\}$

MaxSAT Preprocessing

Definition:
Preprocessing is sound if
$cost(\tau) = cost(\text{REC}(\tau))$
for any optimal solution to $\text{PRE}(\mathcal{F})$.

$\mathcal{F}_H = \{ \quad (x \vee \neg y), (\neg x \vee y) \quad \}$
$\mathcal{F}_S = \{ \quad (\neg x) \quad \}$

$\text{PRE}(\mathcal{F}_H) = \{ \quad (x \vee \neg y) \quad \}$
$\text{PRE}(\mathcal{F}_S) = \{ \quad (\neg x) \quad \}$

$\text{REC}(\tau) = \{\neg x, \neg y\}$
$cost(\text{REC}(\tau)) = 0$

$\tau = \{\neg x, \neg y\}$
$cost(\tau) = 0$

MaxSAT Preprocessing

Take-Home Message 1
Existing proofs of soundness only apply to optimal solutions.

$\mathcal{F}_H = \{ \quad (x \vee \neg y), (\neg x \vee y) \quad \}$
$\mathcal{F}_S = \{ \quad (\neg x) \quad \}$

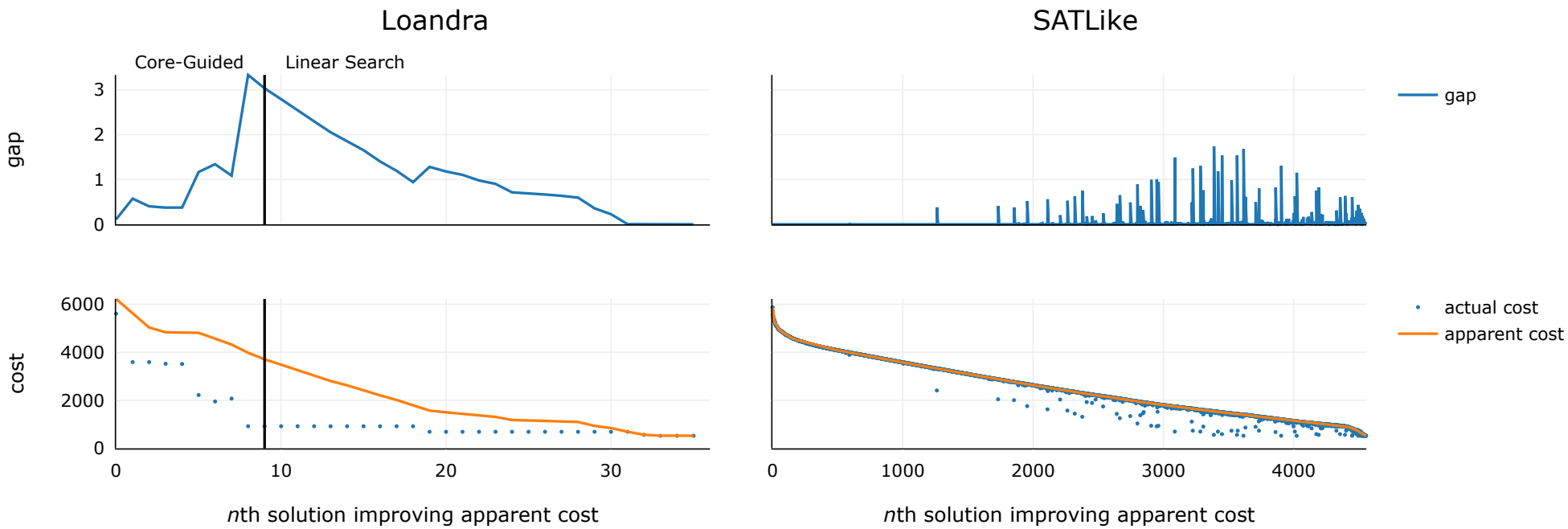
$\textsc{Pre}(\mathcal{F}_H) = \{ \quad (x \vee \neg y) \quad \}$
$\textsc{Pre}(\mathcal{F}_S) = \{ \quad (\neg x) \quad \}$

$\textsc{Rec}(\tau) = \{\neg \mathbf{x}, \neg \mathbf{y}\}$
$\mathit{cost}(\textsc{Rec}(\tau)) = 0$

$\tau = \{\mathbf{x}, \neg \mathbf{y}\}$
$\mathit{cost}(\tau) = 1$

And in practice...

Loandra: Berg et al. [2019], SAT-Like: Lei and Cai [2018]



- Actual Cost $\rightarrow$ $cost(\text{REC}(\tau))$
- Apparent Cost $\rightarrow$ $cost(\tau)$

And in practice...

Loandra: Berg et al. [2019], SAT-Like: Lei and Cai [2018]

Take-Home Message 2

Preprocessing can lead to misinterpreted costs of non-optimal solutions both in theory and practice

And in practice...

Loandra: Berg et al. [2019], SAT-Like: Lei and Cai [2018]

Take-Home Message 2

Preprocessing can lead to misinterpreted costs of non-optimal solutions both in theory and practice

Recall

Non-optimal solutions encountered both in complete and incomplete search

Solution:

Locally Minimal Solutions

Solutions where no falsified soft clause can be ”trivially” satisfied.

Example

We show:

The cost of locally minimal solutions is not misrepresented during search.

Solution:

Locally Minimal Solutions

Solutions where no falsified soft clause can be ”trivially” satisfied.

Example

$\textsc{Pre}(\mathcal{F}_H) = \{(x \vee y)\}$
$\textsc{Pre}(\mathcal{F}_S) = \{(\neg x)\}$
$\tau = \{x, y\}$

We show:

The cost of locally minimal solutions is not misrepresented during search.

Solution:

Locally Minimal Solutions

Solutions where no falsified soft clause can be ”trivially” satisfied.

Example

$\textsc{Pre}(\mathcal{F}_H) = \{(x \vee y)\}$
$\textsc{Pre}(\mathcal{F}_S) = \{(\neg x)\}$
$\tau = \{x, y\}$

$\textsc{Pre}(\mathcal{F}_H) = \{(x \vee y)\}$
$\textsc{Pre}(\mathcal{F}_S) = \{(\neg x)\}$
$\tau_L = \{\neg x, y\}$

We show:

The cost of locally minimal solutions is not misrepresented during search.

Solution:

Locally Minimal Solutions

Solutions where no falsified soft clause can be ”trivially” satisfied.

Example

$\textsc{Pre}(\mathcal{F}_H) = \{(x \vee y)\}$
$\textsc{Pre}(\mathcal{F}_S) = \{(\neg x)\}$
$\tau = \{x, y\}$

$\textsc{Pre}(\mathcal{F}_H) = \{(x \vee y)\}$
$\textsc{Pre}(\mathcal{F}_S) = \{(\neg x)\}$
$\tau_L = \{\neg x, y\}$

We show:

The cost of locally minimal solutions is not misrepresented during search.

More details in the paper

Stochastic Local Search over Locally Minimal Solutions

Key Ideas

- Build on top of the state-of-the-art local search approach SATLike
- Group clauses after preprocessing.
  - put two clauses into the same group if they ”originate” from the same soft clause.
- Direct search toward satisfying groups instead of clauses.
- Greedily minimize solutions obtained during search to guarantee local minimality.

Experiments

Setup

Benchmarks: The 297 weighted instances (from 26 domains) of MaxSAT Evaluation 2019.

Solver variants:

- NO-PREPRO : no preprocessing.
- PREPRO-NOLOCMIN : only preprocessing.
- DEFAULT : preprocessing and searching over locally minimal solutions.

Experiments

Setup

Benchmarks: The 297 weighted instances (from 26 domains) of MaxSAT Evaluation 2019.

Solver variants:

- NO-PREPRO : no preprocessing.
- PREPRO-NOLOCMIN : only preprocessing.
- DEFAULT : preprocessing and searching over locally minimal solutions.

High-level results

DEFAULT $\xrightarrow{\text{wins } \mathbf{23} \text{ domains over}}$ NO-PREPRO $\xrightarrow{\text{wins } \mathbf{15} \text{ domains over}}$ PREPRO-NOLOCMIN

See paper for detailed results!

Experiments

Take-Home Message 3

Searching over locally minimal solutions is crucial for effective use of preprocessing.

Conclusions

In this paper, we:

- show that preprocessing can mislead several different types of complete and incomplete solvers;
- propose locally minimal solutions as a possible fix;
- propose an SLS algorithm that only searches over locally minimal solutions; and
- show that incorporating locally minimal solutions to search improves the empirical performance.

Conclusions

In this paper, we:

- show that preprocessing can mislead several different types of complete and incomplete solvers;
- propose locally minimal solutions as a possible fix;
- propose an SLS algorithm that only searches over locally minimal solutions; and
- show that incorporating locally minimal solutions to search improves the empirical performance.

Conclusions

In this paper, we:

- show that preprocessing can mislead several different types of complete and incomplete solvers;
- propose locally minimal solutions as a possible fix;
- propose an SLS algorithm that only searches over locally minimal solutions; and
- show that incorporating locally minimal solutions to search improves the empirical performance.

Conclusions

In this paper, we:

- show that preprocessing can mislead several different types of complete and incomplete solvers;
- propose locally minimal solutions as a possible fix;
- propose an SLS algorithm that only searches over locally minimal solutions; and
- show that incorporating locally minimal solutions to search improves the empirical performance.

In the future we aim to:

- Use locally minimal solutions to improve complete algorithms.

Bibliography I

F. Bacchus, M. Järvisalo, and R. Martins. MaxSAT Evaluation 2018: New developments and detailed results. *J. Satisfiability, Boolean Modeling and Computation*, 11:99–131, 2019.

J. Berg, E. Demirovic, and P.J. Stuckey. Core-boosted linear search for incomplete MaxSAT. In *Proc. CPAIOR*, volume 11494 of *LNCS*, pages 39–56. Springer, 2019.

Jeremias Berg and Matti Järvisalo. Cost-optimal constrained correlation clustering via weighted partial maximum satisfiability. *Artif. Intell.*, 244:110–142, 2017. URL https://doi.org/10.1016/j.artint.2015.07.001.

Yibin Chen, Sean Safarpour, João Marques-Silva, and Andreas G. Veneris. Automated design debugging with maximum satisfiability. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 29(11):1804–1817, 2010. URL https://doi.org/10.1109/TCAD.2010.2061270.

Emir Demirovic, Nysret Musliu, and Felix Winter. Modeling and solving staff scheduling with partial weighted maxsat. *Annals OR*, 275(1):79–99, 2019. URL https://doi.org/10.1007/s10479-017-2693-y.

Bishwamittra Ghosh and Kuldeep S. Meel. IMLI: an incremental framework for maxsat-based learning of interpretable classification rules. In Vincent Conitzer, Gillian K. Hadfield, and Shannon Vallor, editors, *Proc AIES*, pages 203–210. ACM, 2019. URL https://doi.org/10.1145/3306618.3314283.

Toshinori Hosokawa, Hiroshi Yamazaki, Kenichiro Misawa, Masayoshi Yoshimura, Yuki Hirama, and Masavuki Arai. A low capture power oriented x-filling method using partial maxsat iteratively. In *Proc IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems, DFT*, pages 1–6. IEEE, 2019. URL https://doi.org/10.1109/DFT.2019.8875434.

Z. Lei and S. Cai. Solving (weighted) partial MaxSAT by dynamic local search for SAT. In *Proc. IJCAI*, pages 1346–1352. ijcai.org, 7 2018. doi: 10.24963/ijcai.2018/187. URL https://doi.org/10.24963/ijcai.2018/187.

Lei Zhang and Fahiem Bacchus. MAXSAT heuristics for cost optimal planning. In Jörg Hoffmann and Bart Selman, editors, *Proc AAAI*. AAAI Press, 2012. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5190.