Improving the Effectiveness of SAT-Based Preprocessing for MaxSAT

Jeremias Berg, Paul Saikko and Matti Järvisalo

HIIT & Dept. of Computer Science
University of Helsinki
Finland

February 23, 2018
IJCAI 2015

Motivation

- **(Weighted-Partial) Maximum Satisfiability:** An NP-hard optimization paradigm based on the Boolean Satisfiability problem.
- Example instance:
  $F_h = \{(x \vee z), (\neg z), (y \vee z)\}$
  $F_s = \{(\neg x), (\neg y \vee \neg z), (z \vee y), (\neg z \vee y)\}$
- Find assignment τ that satisfies all hard and a maximum number of soft clauses.
  - Optimize over the weights on soft clauses.
- MaxSAT solving an active area of research.
- Solvers applied for cost-optimal solutions in various domains.

Fu and Malik [2006]
Morgado et al. [2012]
Marques-Silva and Planes [2008]
Morgado et al. [2014]
Ansótegui and Gabàs [2013]
Martins et al. [2014]
Li et al. [2009]
Jose and Majumdar [2011]
Zhu et al. [2011]
Guerra and Lynce [2012]
Heras et al. [2011]
Koshimura et al. [2012]
Davies and Bacchus [2013]
Morgado et al. [2013]
Berg and Järvisalo [2013]
Berg et al. [2014]
Bunte et al. [2014]

Motivation

- **(Weighted-Partial) Maximum Satisfiability:** An NP-hard optimization paradigm based on the Boolean Satisfiability problem.
- Example instance:
  $F_h = \{(x \vee z), (\neg z), (y \vee z)\}$
  $F_s = \{(\neg x), (\neg y \vee \neg z), (z \vee y), (\neg z \vee y)\}$
- Find assignment τ that satisfies all hard and a maximum number of soft clauses.
  - Optimize over the weights on soft clauses.
- MaxSAT solving an active area of research.
- Solvers applied for cost-optimal solutions in various domains.

Fu and Malik [2006]
Morgado et al. [2012]
Marques-Silva and Planes [2008]
Morgado et al. [2014]
Ansótegui and Gabàs [2013]
Martins et al. [2014]
Li et al. [2009]
Jose and Majumdar [2011]
Zhu et al. [2011]
Guerra and Lynce [2012]
Heras et al. [2011]
Koshimura et al. [2012]
Davies and Bacchus [2013]
Morgado et al. [2013]
Berg and Järvisalo [2013]
Berg et al. [2014]
Bunte et al. [2014]

Motivation

- **(Weighted-Partial) Maximum Satisfiability:** An NP-hard optimization paradigm based on the Boolean Satisfiability problem.
- Example instance:
  $F_h = \{(x \vee z), (\neg z), (y \vee z)\}$
  $F_s = \{(\neg x), (\neg y \vee \neg z), (z \vee y), (\neg z \vee y)\}$
- Find assignment τ that satisfies all hard and a maximum number of soft clauses.
  - Optimize over the weights on soft clauses.
- MaxSAT solving an active area of research.
- Solvers applied for cost-optimal solutions in various domains.

Fu and Malik [2006]
Morgado et al. [2012]
Marques-Silva and Planes [2008]
Morgado et al. [2014]
Ansótegui and Gabàs [2013]
Martins et al. [2014]
Li et al. [2009]
Jose and Majumdar [2011]
Zhu et al. [2011]
Guerra and Lynce [2012]
Heras et al. [2011]
Koshimura et al. [2012]
Davies and Bacchus [2013]
Morgado et al. [2013]
Berg and Järvisalo [2013]
Berg et al. [2014]
Bunte et al. [2014]

Motivation

- **(Weighted-Partial) Maximum Satisfiability:** An NP-hard optimization paradigm based on the Boolean Satisfiability problem.
- Example instance:
  $F_h = \{(x \vee z), (\neg z), (y \vee z)\}$
  $F_s = \{(\neg x), (\neg y \vee \neg z), (z \vee y), (\neg z \vee y)\}$
- Find assignment τ that satisfies all hard and a maximum number of soft clauses.
  - Optimize over the weights on soft clauses.
- MaxSAT solving an active area of research.
- Solvers applied for cost-optimal solutions in various domains.

Fu and Malik [2006]
Morgado et al. [2012]
Marques-Silva and Planes [2008]
Morgado et al. [2014]
Ansótegui and Gabàs [2013]
Martins et al. [2014]
Li et al. [2009]
Jose and Majumdar [2011]
Zhu et al. [2011]
Guerra and Lynce [2012]
Heras et al. [2011]
Koshimura et al. [2012]
Davies and Bacchus [2013]
Morgado et al. [2013]
Berg and Järvisalo [2013]
Berg et al. [2014]
Bunte et al. [2014]

Motivation

- **(Weighted-Partial) Maximum Satisfiability:** An NP-hard optimization paradigm based on the Boolean Satisfiability problem.
- Example instance:
  $F_h = \{(x \lor z), (\neg z), (y \lor z)\}$
  $F_s = \{(\neg x), (\neg y \lor \neg z), (z \lor y), (\neg z \lor y)\}$
- Find assignment τ that satisfies all hard and a maximum number of soft clauses.
  - Optimize over the weights on soft clauses.
- MaxSAT solving an active area of research.
- Solvers applied for cost-optimal solutions in various domains.

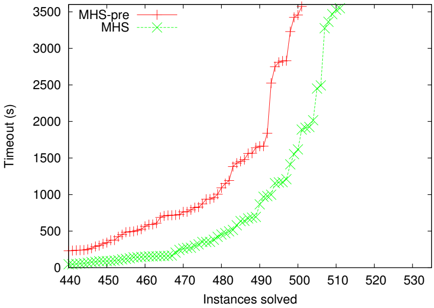
Fu and Malik [2006]
Morgado et al. [2012]
Marques-Silva and Planes [2008]
Morgado et al. [2014]
Ansótegui and Gabàs [2013]
Martins et al. [2014]
Li et al. [2009]
Jose and Majumdar [2011]
Zhu et al. [2011]
Guerra and Lynce [2012]
Heras et al. [2011]
Koshimura et al. [2012]
Davies and Bacchus [2013]
Morgado et al. [2013]
Berg and Järvisalo [2013]
Berg et al. [2014]
Bunte et al. [2014]

SAT-based preprocessing for MaxSAT

- *Preprocessing:* central technique in modern SAT solvers.
  Eén and Biere [2005]; Heule et al. [2010]; Järvisalo et al. [2012]
- Preprocessing MaxSAT has received only little attention.
  - Can be done using the so called Labelled-CNF (LCNF) framework
    Belov and Marques-Silva [2012]
  - Benefits still unclear
  - Can degrade performance of the solver.

SAT-based preprocessing for MaxSAT

- Example: The MaxHS algorithm. Davies and Bacchus [2013]
  One of the best performing solvers of the 2014 MaxSAT evaluation weighted partial crafted category.

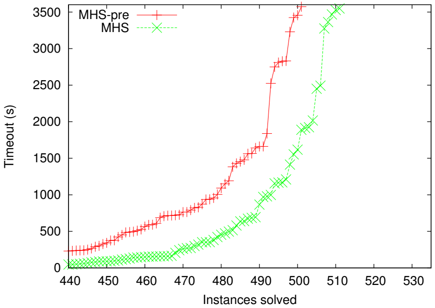


This Work

Focus on improving MaxHS by a tight integration of SAT-based preprocessing.

Main contributions

- Lifting of MaxHS to LCNFs.
- Tighter integration between MaxSAT preprocessing and solving.
- Improves solving time in practice.

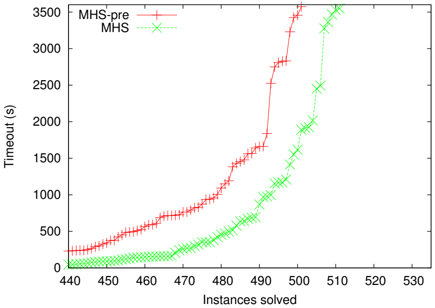


This Work

Focus on improving MaxHS by a tight integration of SAT-based preprocessing.

Main contributions

- Lifting of MaxHS to LCNFs.
- Tighter integration between MaxSAT preprocessing and solving.
- Improves solving time in practice.

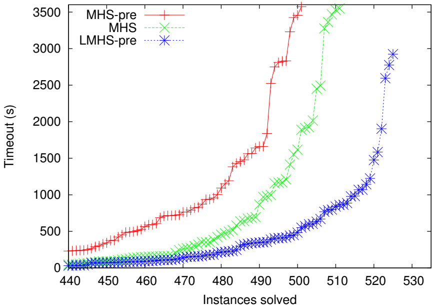


This Work

Focus on improving MaxHS by a tight integration of SAT-based preprocessing.

Main contributions

- Lifting of MaxHS to LCNFs.
- Tighter integration between MaxSAT preprocessing and solving.
- Improves solving time in practice.



Direct application of MaxSAT preprocessing

	MaxSAT	**LCNF**
Soft Clauses	$(x \lor \neg z \lor \neg v)$	
	$(z \lor y)$	
	$(v \lor t)$	
	(t)	
Hard Clauses	$(y \lor t \lor \neg x)$	

Direct application of MaxSAT preprocessing

Conversion to LCNF adds labels to all soft clauses

	MaxSAT		LCNF
Soft Clauses	$(x \lor \neg z \lor \neg v)$	$\longrightarrow$	$(x \lor \neg z \lor \neg v \lor l_1)$
	$(z \lor y)$	$\longrightarrow$	$(z \lor y \lor l_2)$
	$(v \lor t)$	$\longrightarrow$	$(v \lor t \lor l_3)$
	(t)	$\longrightarrow$	$(t \lor l_4)$
Hard Clauses	$(y \lor t \lor \neg x)$	$\longrightarrow$	$(y \lor t \lor \neg x)$

Direct application of MaxSAT preprocessing

Conversion to LCNF adds labels to all soft clauses

MaxSAT Preprocessing can mix labels **LCNF**

Soft Clauses

$(x \lor y \lor t \lor l_1 \lor l_2 \lor l_3)$

$(t \lor l_4)$

Hard Clauses

$(y \lor t \lor \neg x)$

Direct application of MaxSAT preprocessing

Conversion to MaxSAT adds new variables and clauses

MaxSAT		LCNF
Soft Clauses	$(\neg l_1)$	$(x \lor y \lor t \lor l_1 \lor l_2 \lor l_3)$
	$(\neg l_2)$	
	$(\neg l_3)$	
	$(\neg l_4)$	$(t \lor l_4)$
Hard Clauses	$(x \lor y \lor t \lor l_1 \lor l_2 \lor l_3)$	
	$(t \lor l_4)$	
	$(y \lor t \lor \neg x)$	$(y \lor t \lor \neg x)$

Direct application of MaxSAT preprocessing

MaxSAT | | **LCNF**

Soft Clauses

$\cancel{(\neg l_1)}$ $(\neg l_1 \lor a_1)$

$\cancel{(\neg l_2)}$ $(\neg l_2 \lor a_2)$

$\cancel{(\neg l_3)}$ $(\neg l_3 \lor a_3)$

$\cancel{(\neg l_4)}$ $(\neg l_4 \lor a_4)$

MaxHS controls soft clauses by adding assumption variables

Hard Clauses

$(x \lor y \lor t \lor l_1 \lor l_2 \lor l_3)$

$(t \lor l_4)$

$(y \lor t \lor \neg x)$

Direct application of MaxSAT preprocessing

MaxSAT | **LCNF**

Soft Clauses

$\cancel{(\neg l_1)}$ $\cancel{(\neg l_1 \vee a_1)}$

$\cancel{(\neg l_2)}$ $\cancel{(\neg l_2 \vee a_2)}$

$\cancel{(\neg l_3)}$ $\cancel{(\neg l_3 \vee a_3)}$

$\cancel{(\neg l_4)}$ $\cancel{(\neg l_4 \vee a_4)}$

Hard Clauses

$(x \vee y \vee t \vee \mathbf{l_1} \vee \mathbf{l_2} \vee \mathbf{l_3})$

$(t \vee \mathbf{l_4})$

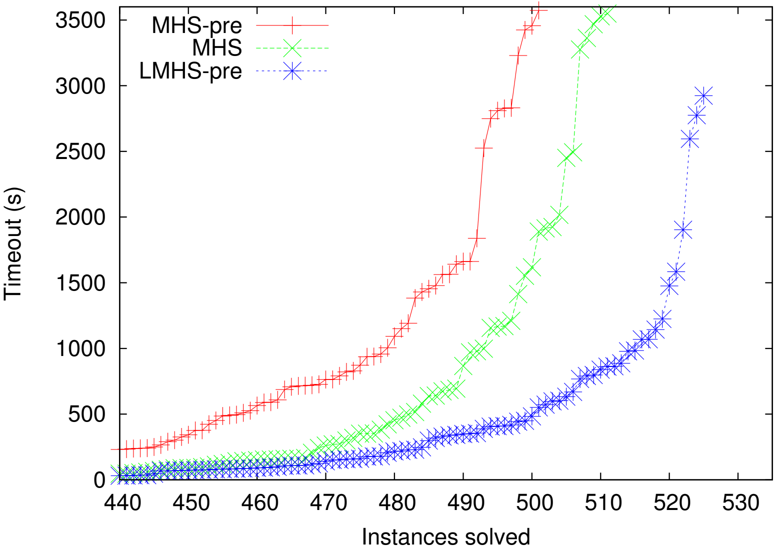
$(y \vee t \vee \neg x)$

Idea of Lifting
Reuse variables
introduced in
LCNF conversion
as assumptions

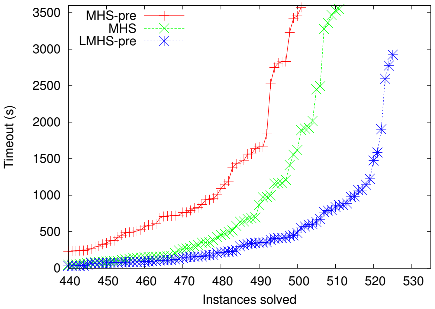
Lifting of MaxHS to LCNF:s

- Label centric reformulation of the MaxHS algorithm
  Davies and Bacchus [2013]
  - No extra assumption variables added during solving
    - Explicit list of assumption variables as input.
  - More than one assumption per clause allowed.

Experimental Evaluation



Conclusions



- Preprocessing MaxSAT can be done using LCNFs.
- Direct application degrades performance of the MaxHS algorithm.
- We propose a lifting of MaxHS to LCNF formulas.
- Tighter integration between MaxSAT preprocessing and solving.
- Improves performance over the original MaxHS algorithm.

Bibliography I

C. Ansótegui and J. Gabàs. Solving (weighted) partial MaxSAT with ILP. In *Proc. CPAIOR*, volume 7874 of *LNCS*, pages 403–409. Springer, 2013.

A. Belov and J. Marques-Silva. Generalizing redundancy in propositional logic: Foundations and hitting sets duality. *CoRR*, abs/1207.1257, 2012. URL `http://arxiv.org/abs/1207.1257`.

J. Berg and M. Järvisalo. Optimal correlation clustering via MaxSAT. In *Proc. 2013 IEEE ICDM Workshops*, pages 750–757. IEEE Press, 2013.

J. Berg, M. Järvisalo, and B. Malone. Learning optimal bounded treewidth bayesian networks via maximum satisfiability. In *Proc. AISTATS*, volume 33, pages 86–95. JMLR, 2014.

Kerstin Bunte, Matti Järvisalo, Jeremias Berg, Petri Myllymäki, Jaakko Peltonen, and Samuel Kaski. Optimal neighborhood preserving visualization by maximum satisfiability. In *Proc. AAAI*, pages 1694–1700. AAAI Press, 2014.

J. Davies and F. Bacchus. Postponing optimization to speed up MAXSAT solving. In *Proc. CP*, volume 8124 of *LNCS*, pages 247–262. Springer, 2013.

N. Eén and A. Biere. Effective preprocessing in SAT through variable and clause elimination. In *Proc. SAT*, volume 3569 of *LNCS*, pages 61–75. Springer, 2005.

Zhaohui Fu and Sharad Malik. On solving the partial MAX-SAT problem. In *Proc. SAT*, volume 4121 of *LNCS*, pages 252–265. Springer, 2006.

J. Guerra and I. Lynce. Reasoning over biological networks using maximum satisfiability. In *Proc. CP*, volume 7514 of *LNCS*, pages 941–956. Springer, 2012.

F. Heras, A. Morgado, and J. Marques-Silva. Core-guided binary search algorithms for maximum satisfiability. In *Proc. AAAI*. AAAI Press, 2011.

M. Heule, M. Järvisalo, and A. Biere. Clause elimination procedures for CNF formulas. In *Proc. LPAR-17*, volume 6397 of *LNCS*, pages 357–371. Springer, 2010.

Matti Järvisalo, Marijn Heule, and Armin Biere. Inprocessing rules. In *Proc. IJCAR*, volume 7364 of *LNCS*, pages 355–370. Springer, 2012.

M. Jose and R. Majumdar. Cause clue clauses: error localization using maximum satisfiability. In *Proc. PLDI*, pages 437–446. ACM, 2011.

Bibliography II

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.

Chu Min Li, Felip Manyà, Nouredine Ould Mohamedou, and Jordi Planes. Exploiting cycle structures in Max-SAT. In *Proc. SAT*, volume 5584 of *LNCS*, pages 467–480. Springer, 2009.

João Marques-Silva and Jordi Planes. Algorithms for maximum satisfiability using unsatisfiable cores. In *Proc. DATE*, pages 408–413. IEEE, 2008.

R. Martins, S. Joshi, V.M. Manquinho, and I. Lynce. Incremental cardinality constraints for MaxSAT. In *Proc. CP*, volume 8656 of *LNCS*, pages 531–548. Springer, 2014.

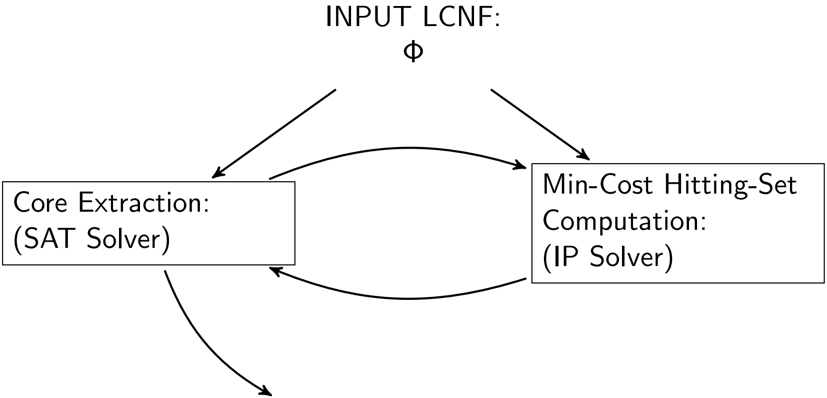
A. Morgado, F. Heras, M.H. Liffiton, J. Planes, and J. Marques-Silva. Iterative and core-guided MaxSAT solving: A survey and assessment. *Constraints*, 18(4):478–534, 2013.

A. Morgado, C. Dodaro, and J. Marques-Silva. Core-guided maxsat with soft cardinality constraints. In *Proc. CP*, volume 8656 of *LNCS*, pages 564–573. Springer, 2014.

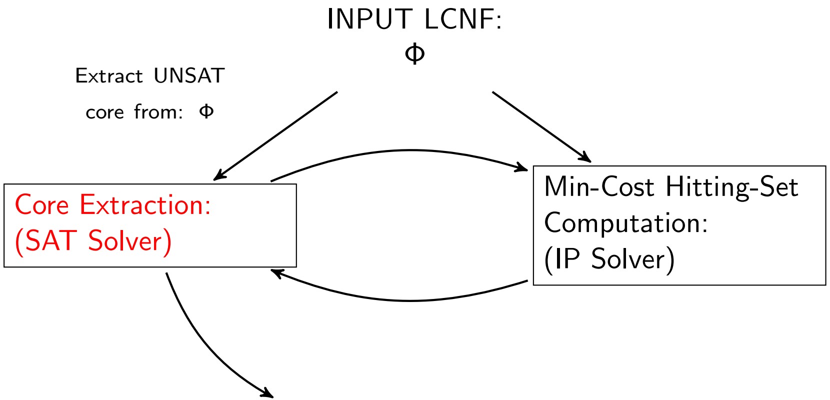
António Morgado, Federico Heras, and João Marques-Silva. Improvements to core-guided binary search for MaxSAT. In *Proc. SAT*, volume 7317 of *LNCS*, pages 284–297. Springer, 2012. ISBN 978-3-642-31611-1.

C.S. Zhu, G. Weissenbacher, and S. Malik. Post-silicon fault localisation using maximum satisfiability and backbones. In *Proc. FMCAD*, pages 63–66, 2011.

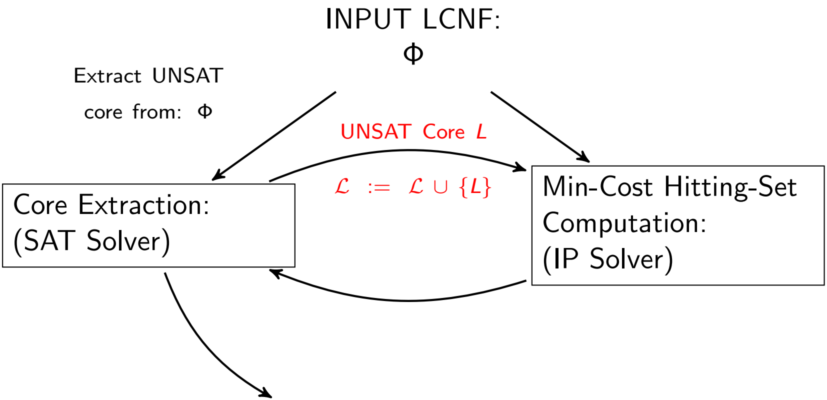
LCNF-MaxHS


INPUT LCNF:
Φ
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

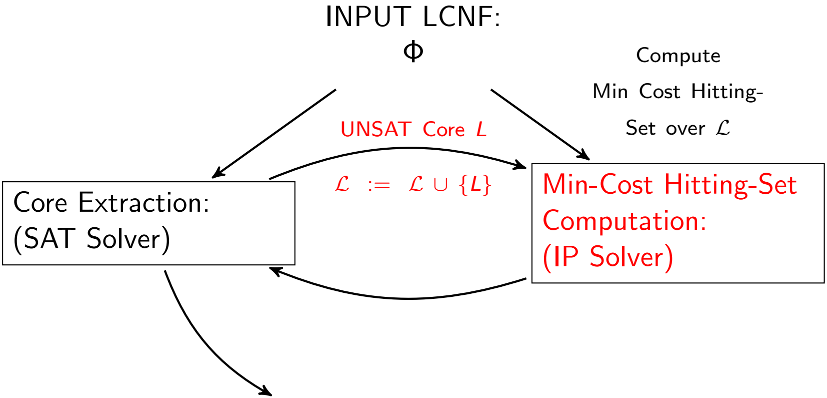
LCNF-MaxHS


INPUT LCNF:
Φ
Extract UNSAT
core from: Φ
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

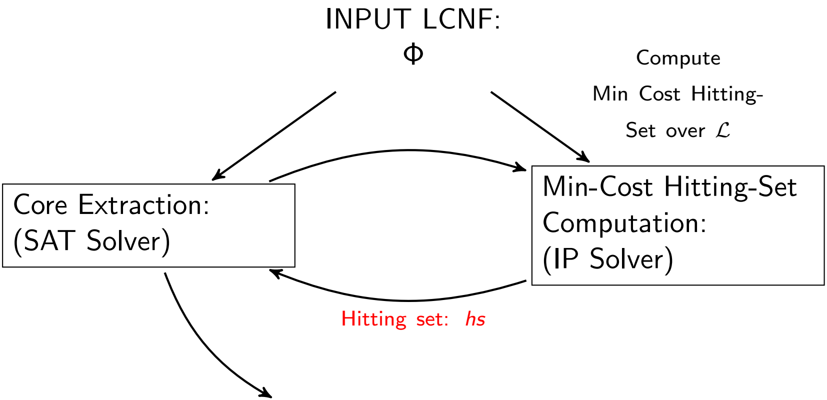
LCNF-MaxHS


INPUT LCNF:
Φ
Extract UNSAT
core from: Φ
UNSAT Core L
$\mathcal{L} := \mathcal{L} \cup \{L\}$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

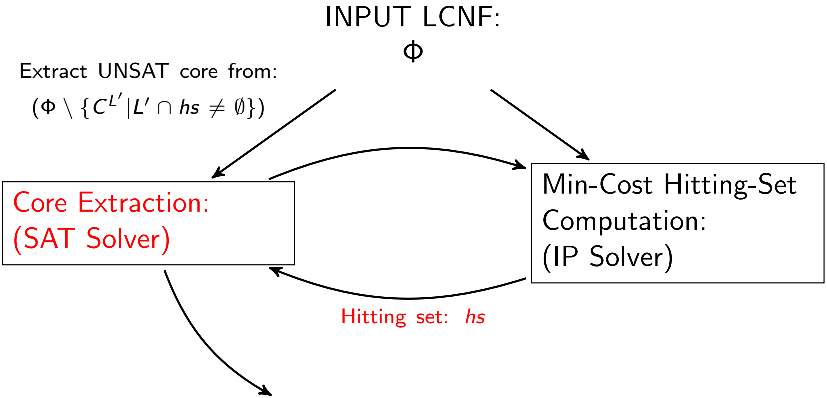
LCNF-MaxHS


INPUT LCNF:
Φ
Compute
Min Cost Hitting-
Set over $\mathcal{L}$
UNSAT Core L
$\mathcal{L} := \mathcal{L} \cup \{L\}$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

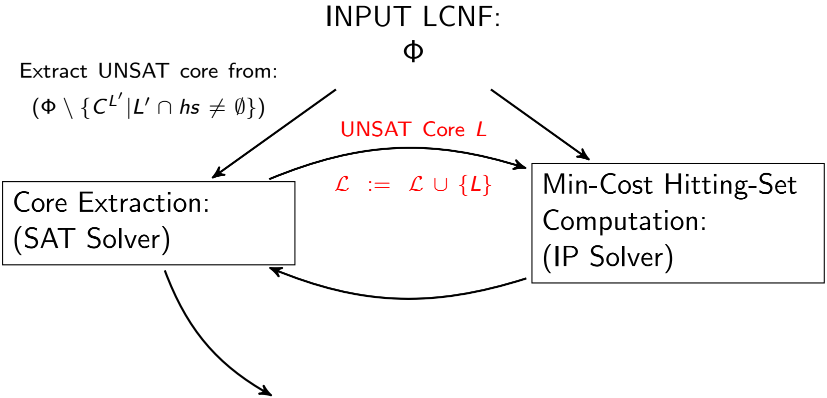
LCNF-MaxHS


INPUT LCNF:
Φ
Compute
Min Cost Hitting-
Set over $\mathcal{L}$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)
Hitting set: *hs*

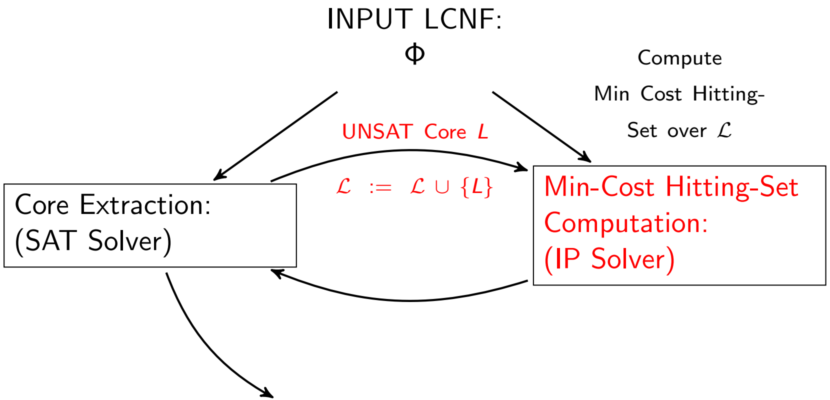
LCNF-MaxHS


INPUT LCNF:
Φ
Extract UNSAT core from:
$(\Phi \setminus \{C^{L'} | L' \cap hs \neq \emptyset\})$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)
Hitting set: hs

LCNF-MaxHS

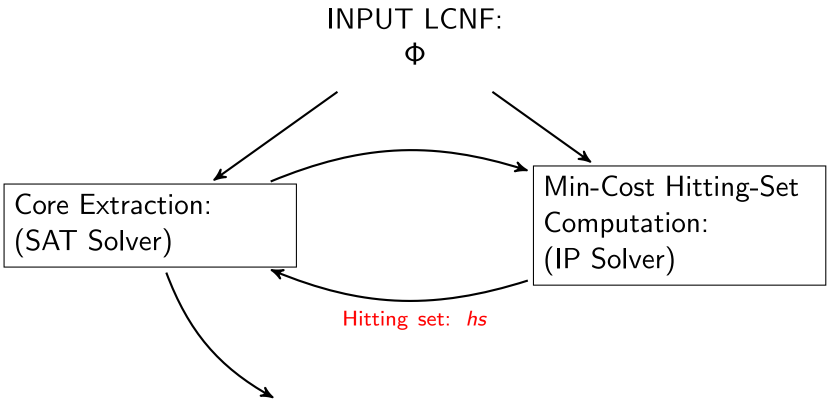

INPUT LCNF:
Φ
Extract UNSAT core from:
$(\Phi \setminus \{C^{L'} \mid L' \cap hs \neq \emptyset\})$
UNSAT Core L
$\mathcal{L} := \mathcal{L} \cup \{L\}$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

LCNF-MaxHS

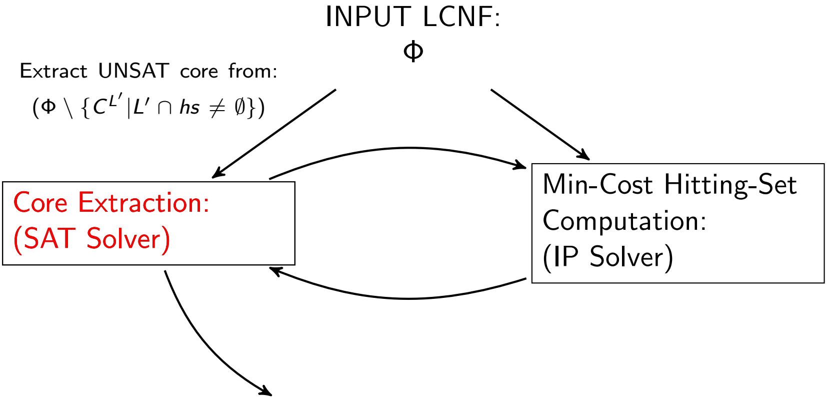


INPUT LCNF:
Φ
Compute
Min Cost Hitting-
Set over $\mathcal{L}$
UNSAT Core L
$\mathcal{L} := \mathcal{L} \cup \{L\}$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

LCNF-MaxHS

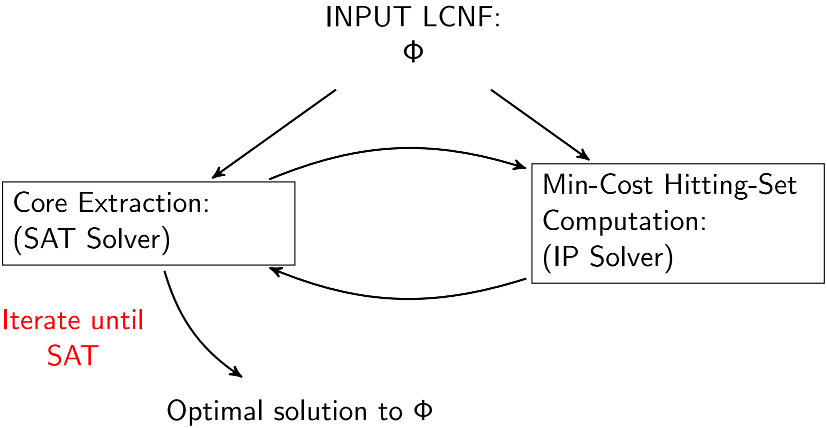

INPUT LCNF:
Φ
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)
Hitting set: *hs*

LCNF-MaxHS



INPUT LCNF:
Φ
Extract UNSAT core from:
$(\Phi \setminus \{C^{L'} \mid L' \cap hs \neq \emptyset\})$
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)

LCNF-MaxHS


INPUT LCNF:
Φ
Core Extraction:
(SAT Solver)
Min-Cost Hitting-Set
Computation:
(IP Solver)
Iterate until
SAT
Optimal solution to Φ