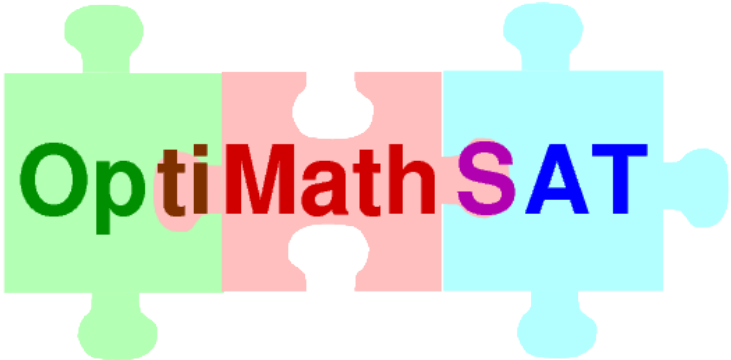
On the Benefits of Enhancing Optimization Modulo Theories with Sorting Networks for MaxSMT

Roberto Sebastiani, Patrick Trentin
roberto.sebastiani@unitn.it
trentin@disi.unitn.it

DISI, University of Trento



SMT Workshop, July 1st, 2016

Contents

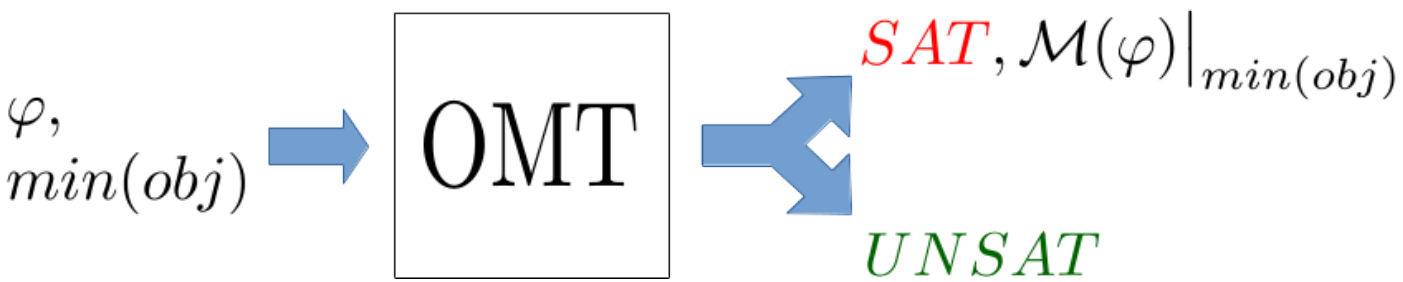
Contents

Optimization Modulo Theories (OMT) [15, 16, 18]

Optimization Modulo Theories with $\mathcal{LA}$ objectives [15, 16, 6, 7, 17, 13, 4, 5]

problem of finding a model for $\langle \varphi, obj \rangle$, minimizing the value of *obj*:

- *obj* being a $\mathcal{LA}$ or Pseudo-Boolean cost function
- maximization dual
- extended to multiple objectives (linear combination, min-max, boxed, lexicographic, Pareto) [13, 4, 5, 20, 19]
- incremental [5, 20]



OMT Applications

Formal Verification

- Formal **Verification** of parametric systems [18, 14]
- SW verification & synthesis [12, 14]
  (e.g. BMC, invariant generation, program syntehsis, ...)
- computation of worst-case execution time of loop-free programs [11]
- computation of optimal structure of undirected Markov network [9]
- computation abstract transformers [13]
- ...

OMT Applications

Formal Verification

- Formal **Verification** of parametric systems [18, 14]
- SW verification & synthesis [12, 14]
  (e.g. BMC, invariant generation, program syntehsis, ...)
- computation of worst-case execution time of loop-free programs [11]
- computation of optimal structure of undirected Markov network [9]
- computation abstract transformers [13]
- ...

Other

Machine Learning PYLMT [22], a Structured Learning Modulo Theories [22] tool that performs inference and learning in hybrid domains

Requirement Engineering CGM-TOOL [1], a tool for computing optimal realization of a Goal Model enriched with preferences and resources

Partial Weighted MaxSMT [15, 6, 7, 16, 17]

A pair $\langle \varphi_h, \varphi_s \rangle$, where

- φ_h: set of "hard" $\mathcal{T}$-clauses
- φ_s: set of *positive-weighted* "soft" $\mathcal{T}$-clauses

goal: find ψ, $\psi \subseteq \varphi_s$, s.t. $\varphi_h \cup \psi$ is $\mathcal{T}$-satisfiable and ψ has maximum-weight

Partial Weighted MaxSMT [15, 6, 7, 16, 17]

A pair $\langle \varphi_h, \varphi_s \rangle$, where

- φ_h: set of "hard" $\mathcal{T}$-clauses
- φ_s: set of *positive-weighted* "soft" $\mathcal{T}$-clauses

goal: find ψ, $\psi \subseteq \varphi_s$, s.t. $\varphi_h \cup \psi$ is $\mathcal{T}$-satisfiable and ψ has maximum-weight

Approaches

- MaxSAT engine + SMT's $\mathcal{T}$-Solvers
- encoded as OMT Pseudo-Boolean objective

Partial Weighted MaxSMT [15, 6, 7, 16, 17]

A pair $\langle \varphi_h, \varphi_s \rangle$, where

- φ_h: set of "hard" $\mathcal{T}$-clauses
- φ_s: set of *positive-weighted* "soft" $\mathcal{T}$-clauses

goal: find ψ, $\psi \subseteq \varphi_s$, s.t. $\varphi_h \cup \psi$ is $\mathcal{T}$-satisfiable and ψ has maximum-weight

Approaches

- MaxSAT engine + SMT's $\mathcal{T}$-Solvers
- encoded as OMT Pseudo-Boolean objective

SMT + MaxSAT engine

++ very efficient
(for *pure* MaxSMT)

OMT encoding:

++ can be used when *objective* is given by the **linear (or min-max) combination** of **Pseudo-Boolean** and **Arithmetic terms** (e.g. LGDP [17], or [22]).

++ can handle multiple objectives at the same time as in [20]

OMT encoding

Given $\langle \varphi_h, \varphi_s \rangle$, for each $C_i \in \varphi_s$ introduce fresh Boolean variable A_i

$$\varphi \triangleq \varphi_h \cup \bigcup_{C_i \in \varphi_s} \{(A_i \vee C_i)\}; \;\; obj \triangleq \sum_{C_i \in \varphi_s} w_i A_i \tag{1}$$

its **OMT encoding** is a pair $\langle \varphi, obj \rangle$ [17]

$$\begin{aligned} \varphi &\stackrel{\text{def}}{=} \varphi_h \wedge \textstyle\bigwedge_i ((A_i \rightarrow (x_i = w_i)) \wedge (\neg A_i \rightarrow (x_i = 0))) \wedge \\ &\quad \textstyle\bigwedge_i ((0 \leq x_i) \wedge (x_i \leq w_i))^* \\ obj &\stackrel{\text{def}}{=} \textstyle\sum_i x_i, \;\; x_i \text{ fresh Real variable} \end{aligned}$$

***:** Term $\bigwedge_i \ldots$ + *Early Pruning* = improved efficiency

OMT encoding

Given $\langle \varphi_h, \varphi_s \rangle$, for each $C_i \in \varphi_s$ introduce fresh Boolean variable A_i

$$\varphi \triangleq \varphi_h \cup \bigcup_{C_i \in \varphi_s} \{(A_i \vee C_i)\}; \;\; obj \triangleq \sum_{C_i \in \varphi_s} w_i A_i \tag{1}$$

its **OMT encoding** is a pair $\langle \varphi, obj \rangle$ [17]

$$\begin{aligned} \varphi \;\; &\stackrel{\text{def}}{=} \;\; \varphi_h \wedge \textstyle\bigwedge_i ((A_i \rightarrow (x_i = w_i)) \wedge (\neg A_i \rightarrow (x_i = 0))) \wedge \\ &\quad\;\; \textstyle\bigwedge_i ((0 \leq x_i) \wedge (x_i \leq w_i))^* \\ obj \;\; &\stackrel{\text{def}}{=} \;\; \textstyle\sum_i x_i, \;\; x_i \text{ fresh Real variable} \end{aligned}$$

***:** Term $\bigwedge_i \ldots$ + *Early Pruning* = improved efficiency

Problem:

-- **Performance bottleneck** when dealing with **Pseudo-Boolean objectives** in the form

$$w_1 \cdot \sum_i A_i + \ldots + w_n \cdot \sum_j A_j$$

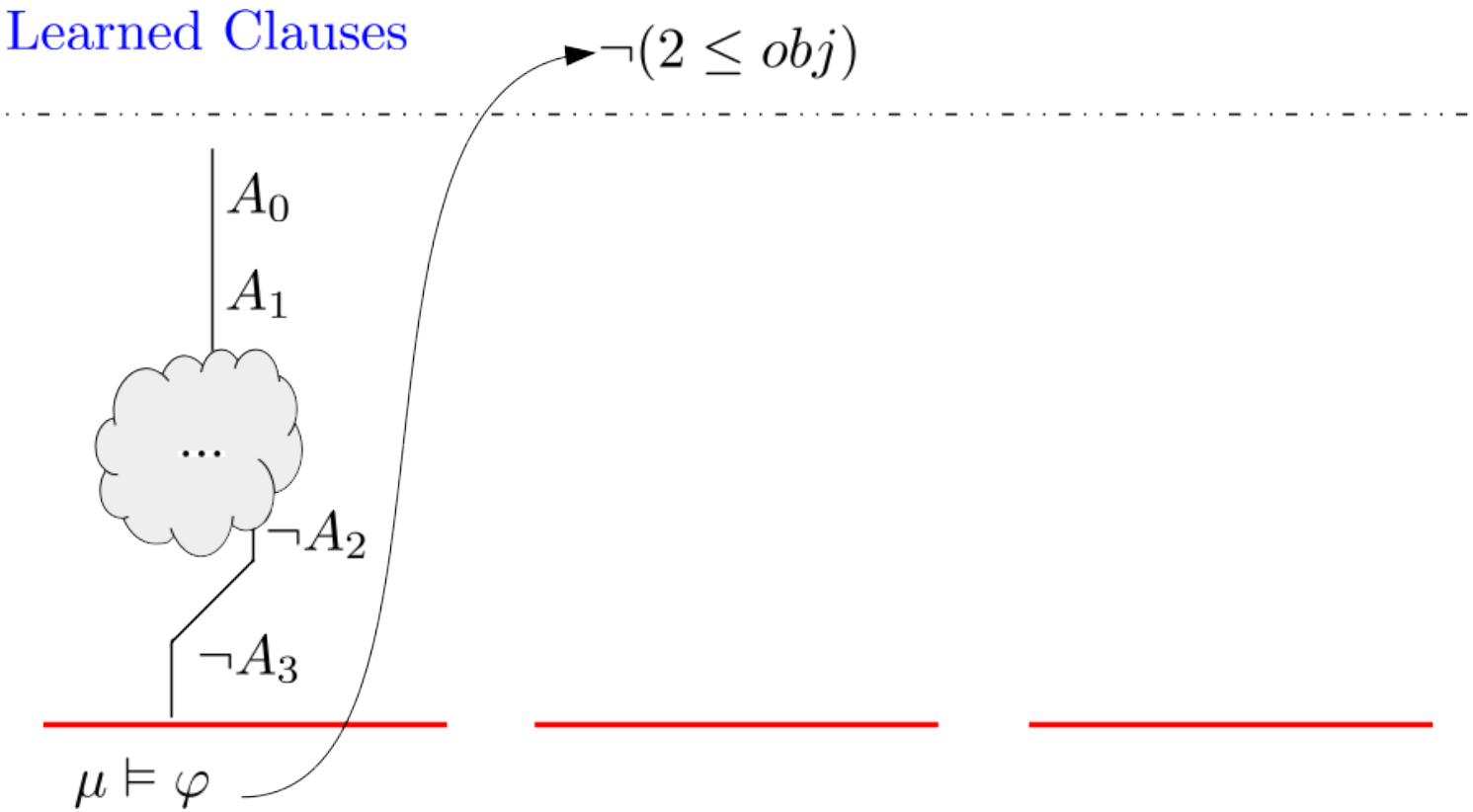
Contents

Running Example: efficiency issue

Problem:

- $\langle \varphi, \texttt{min}(obj) \rangle$, where $obj := w \cdot \sum_{i=0}^{n-1} A_i$, currently $obj = k \cdot w$
- OPTIMIZATION STEP: learn $\neg(k \cdot w \leq obj)$ and restart/jump to *level* 0

Example: with $k = 2$, $w = 1$ and $n = 4$

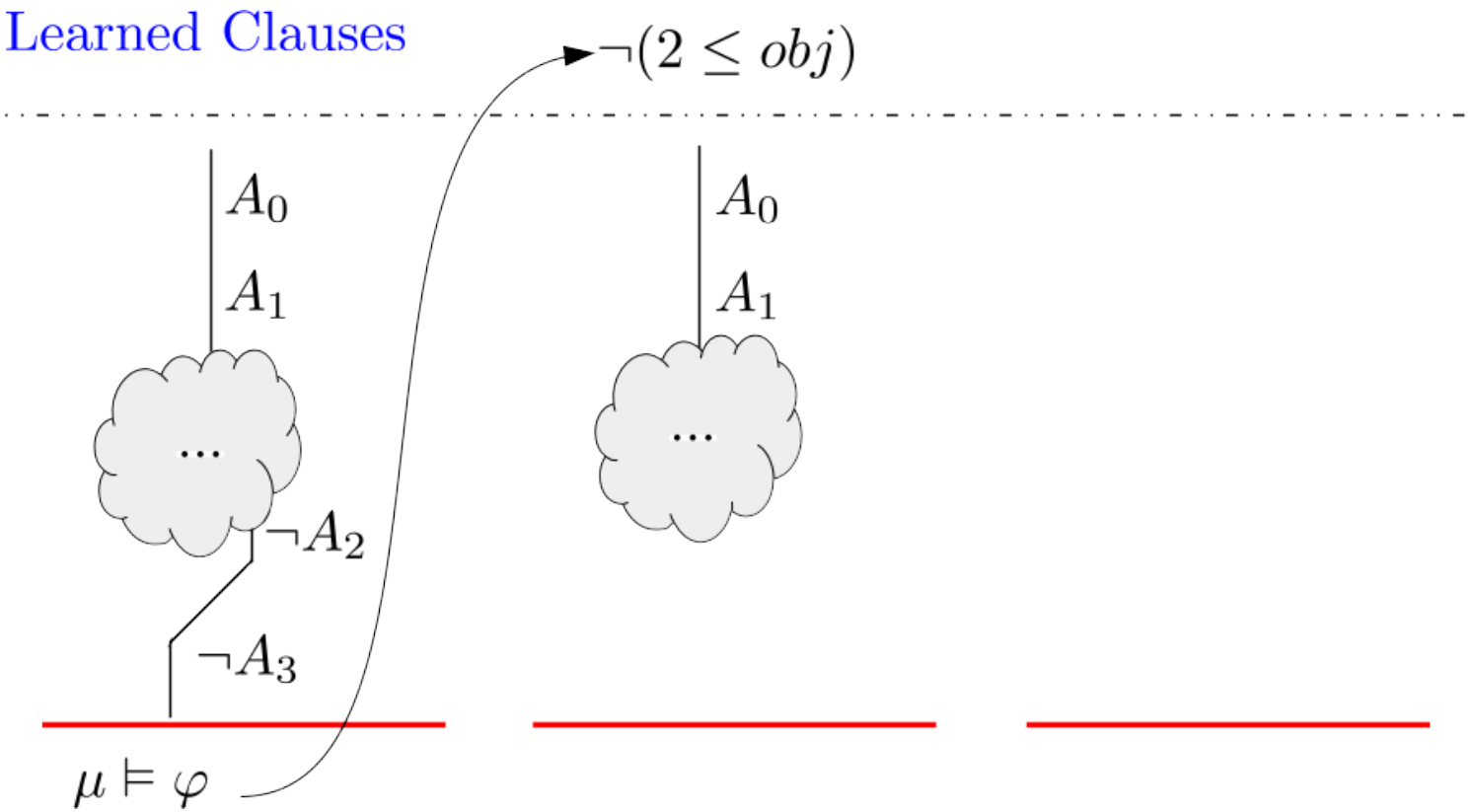


Running Example: efficiency issue

Problem:

- $\neg(k \leq obj)$ causes the inconsistency of $\binom{n}{k}$ truth assignments satisfying exactly k variables in $A_0, ..., A_{n-1}$

Example: with $k = 2$, $w = 1$ and $n = 4$



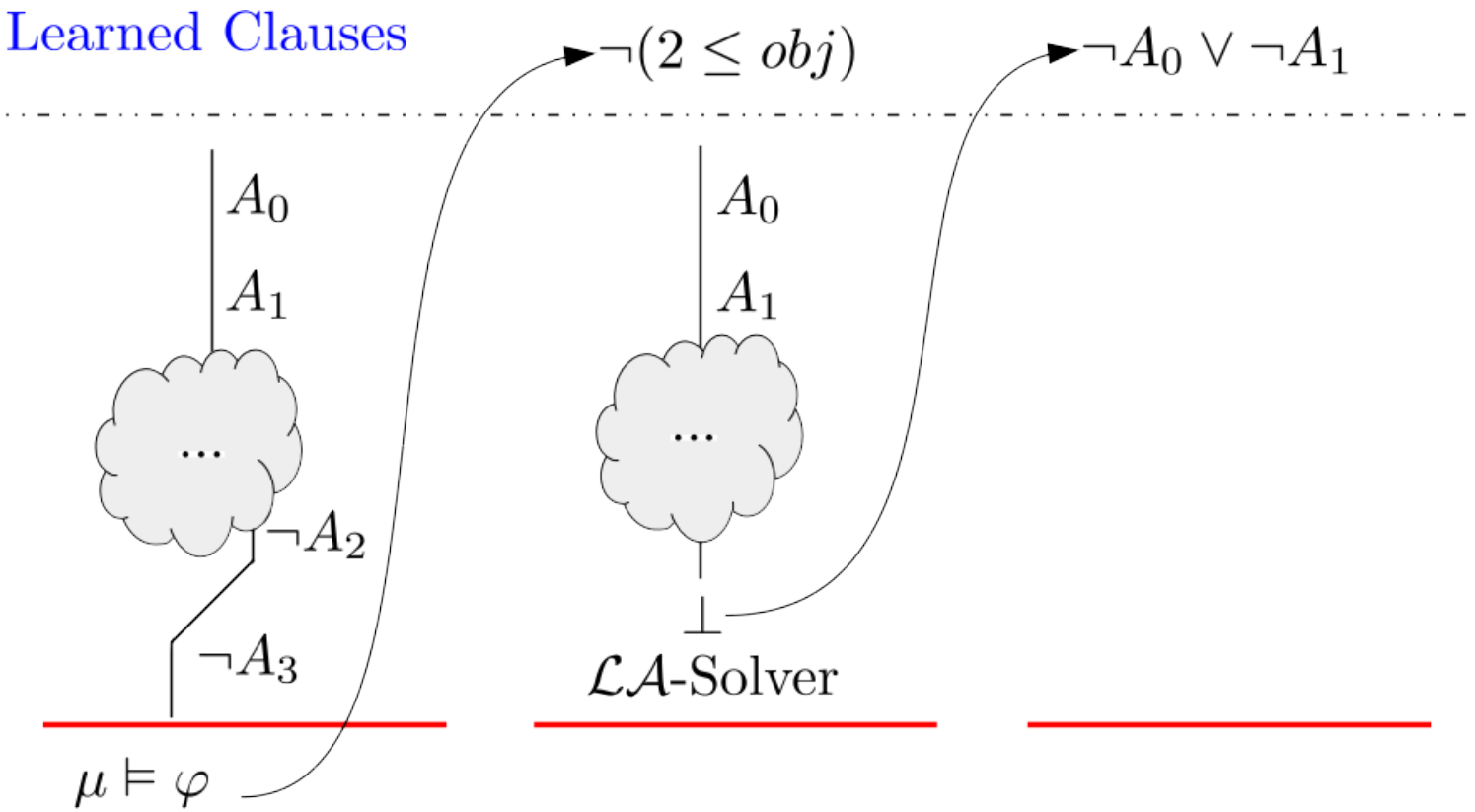
Running Example: efficiency issue

Problem:

- $\neg(k \leq obj)$ causes the inconsistency of $\binom{n}{k}$ truth assignments satisfying exactly k variables in $A_0, ..., A_{n-1}$

  $\Longrightarrow$ inconsistency is not revealed by Boolean Propagation

Example: with $k = 2$, $w = 1$ and $n = 4$

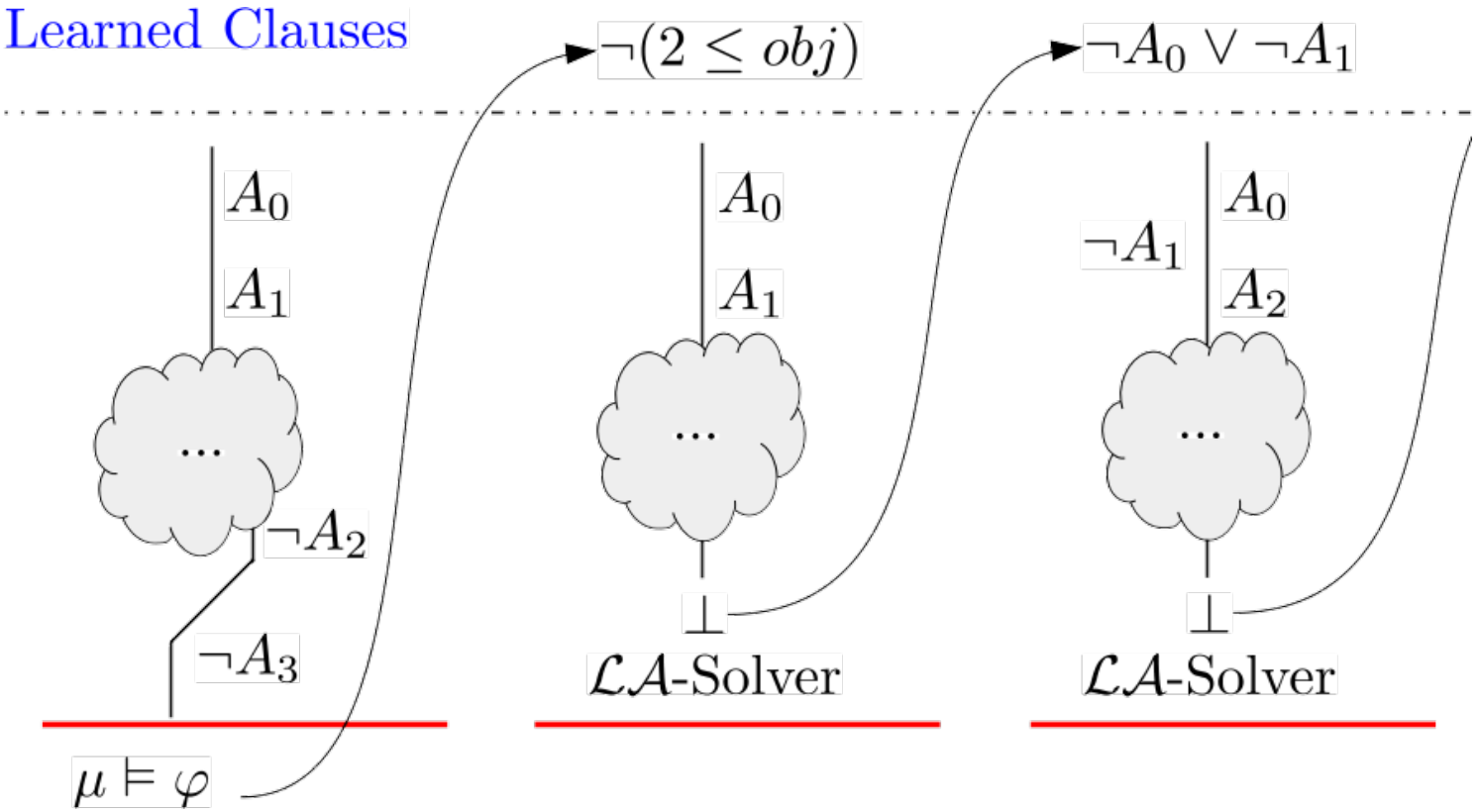


Running Example: efficiency issue

Problem:

- up to $\binom{n}{k}$ (expensive) calls to the $\mathcal{LA}$-Solver required

Example: with $k = 2$, $w = 1$ and $n = 4$



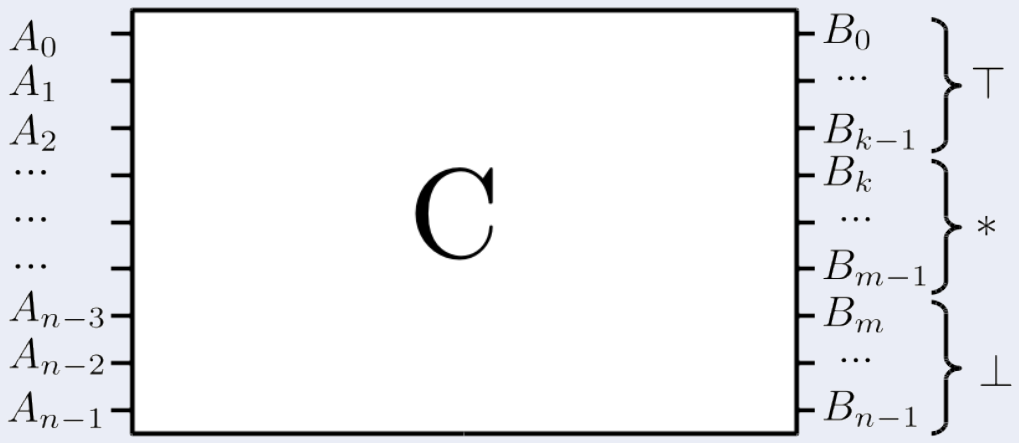
Contents

Solution: Combine OMT with Sorting Networks

Idea. enrich encoding with bi-directional **sorting networks** [21, 10, 3, 2]

Given $\langle \varphi, obj \rangle$, $obj := w \cdot \sum_{i=0}^{n-1} A_i$, and a **sorting network** relation $C(A_0, ..., A_{n-1}, B_0, ..., B_{n-1})$ s.t.

- k A_i's are $\top \iff$ $\{B_0, ..., B_{k-1}\}$ are $\top$,
- $m - k$ A_i's are $* \iff$ $\{B_k, ..., B_{m-1}\}$ are $*$,
- $n - m$ A_i's are $\bot \iff$ $\{B_m, ..., B_{n-1}\}$ are $\bot$



then we encode it as $\langle \varphi', obj \rangle$, where

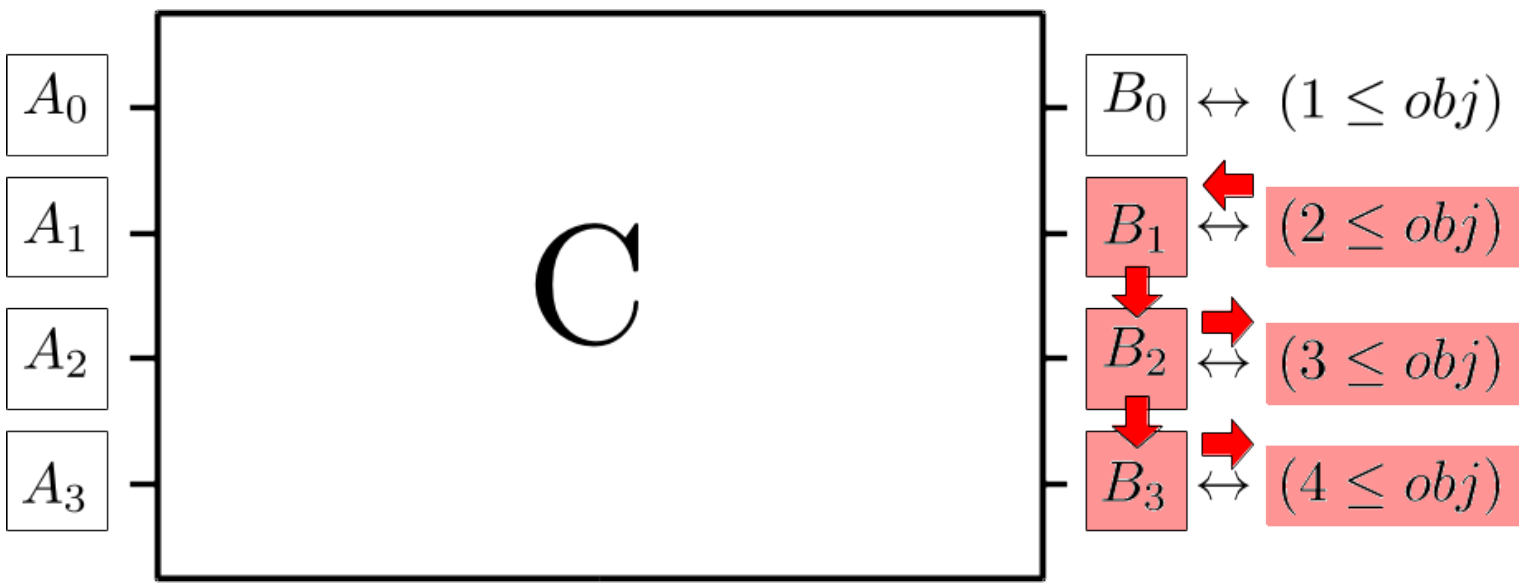
$$\varphi' := \varphi \wedge C(A_0, ..., A_{n-1}) \wedge \bigwedge_{i=0}^{n-1} B_i \leftrightarrow ((k+1) \cdot w \leq obj) \wedge \bigwedge_{i=0}^{n-2} B_{i+1} \rightarrow B_i$$

OMT with Sorting Network Relation

Properties:

- if $(k \cdot w \leq obj) = \bot$, then by BCP $\forall i \in [k, n].B_{i-1} = \bot$

Example: with $k = 2$, $w = 1$ and $n = 4$



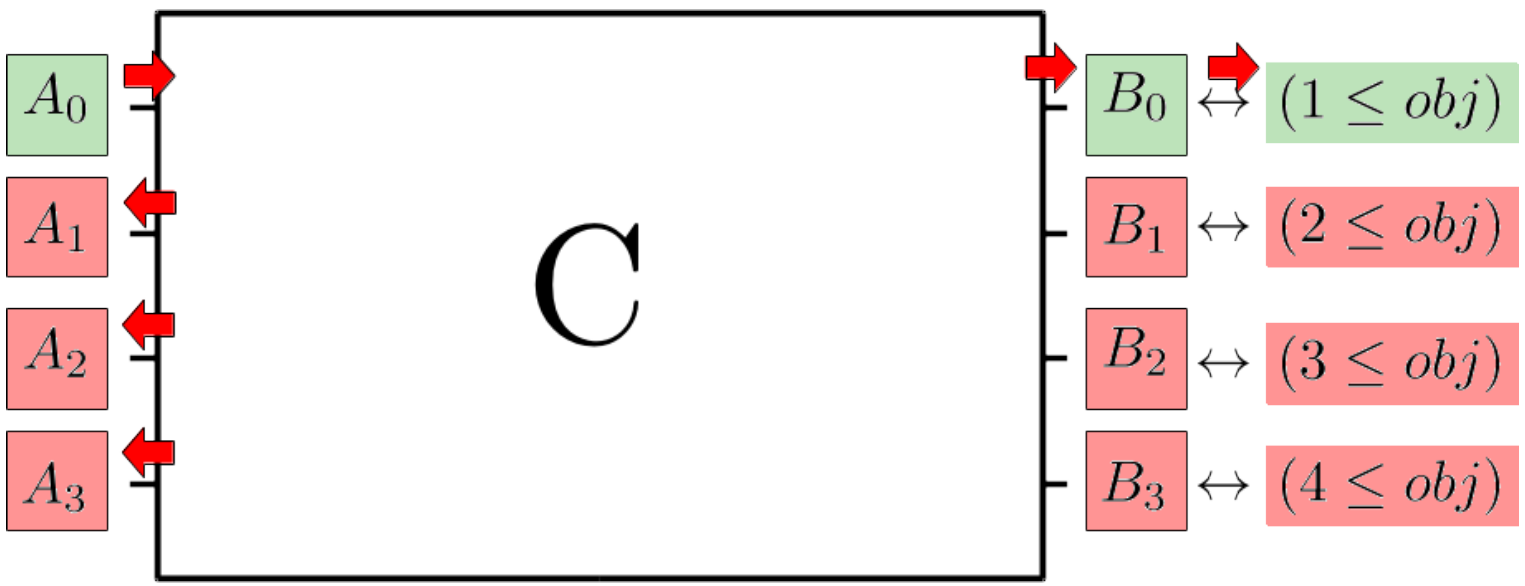
OMT with Sorting Network Relation

Properties:

- if $(k \cdot w \leq obj) = \bot$, then by BCP $\forall i \in [k, n].B_{i-1} = \bot$
- as soon as $k - 1$ A_i are assigned $\top$
  $\Longrightarrow$ all others are unit-propagated to $\bot$

Dual if $(k \cdot w \leq obj) = \top$.

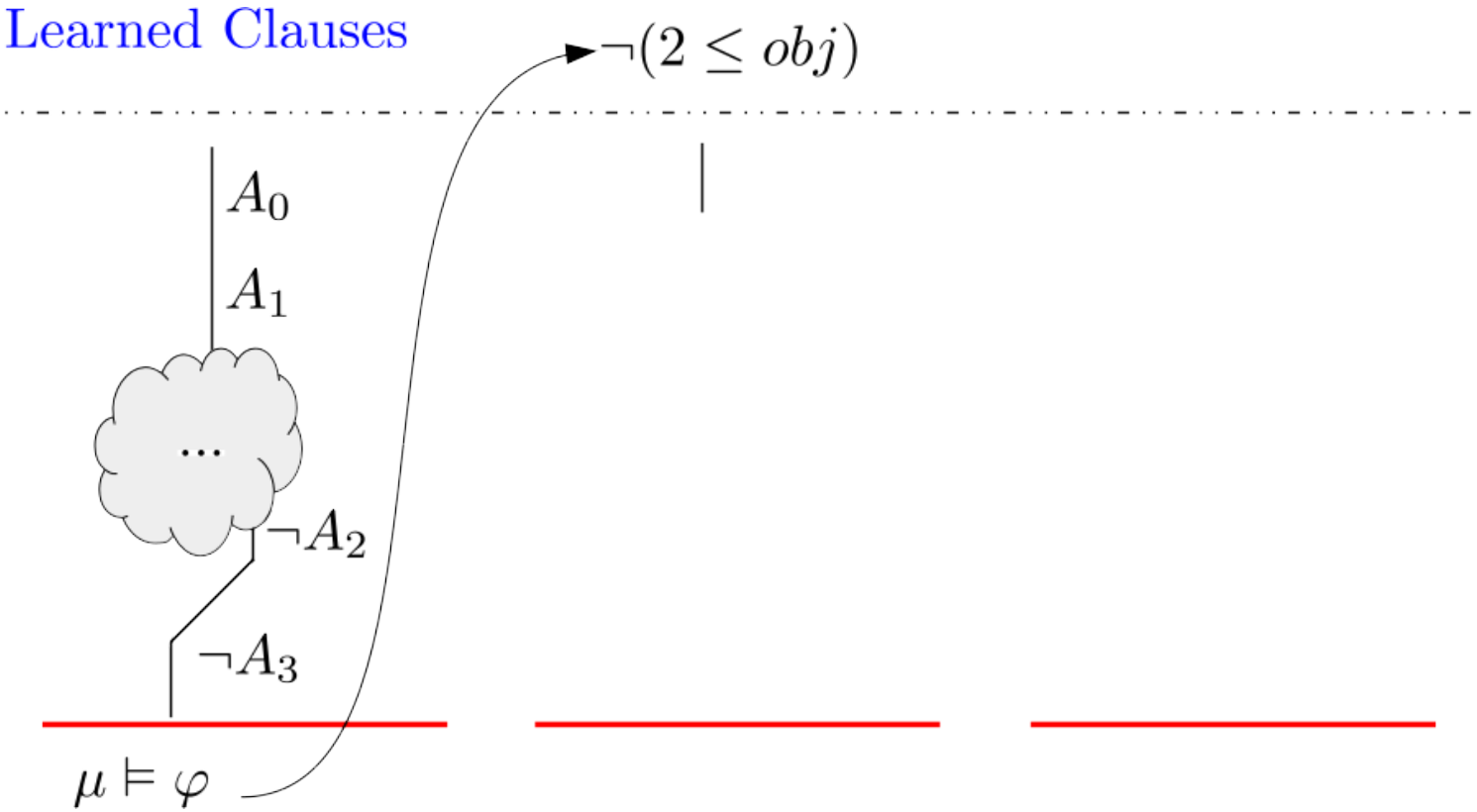
Example: with $k = 2$, $w = 1$ and $n = 4$



Running Example: OMT with sorting networks

- OPTIMIZATION STEP: learn $\neg(k \cdot w \leq obj)$ and restart/jump to *level* 0

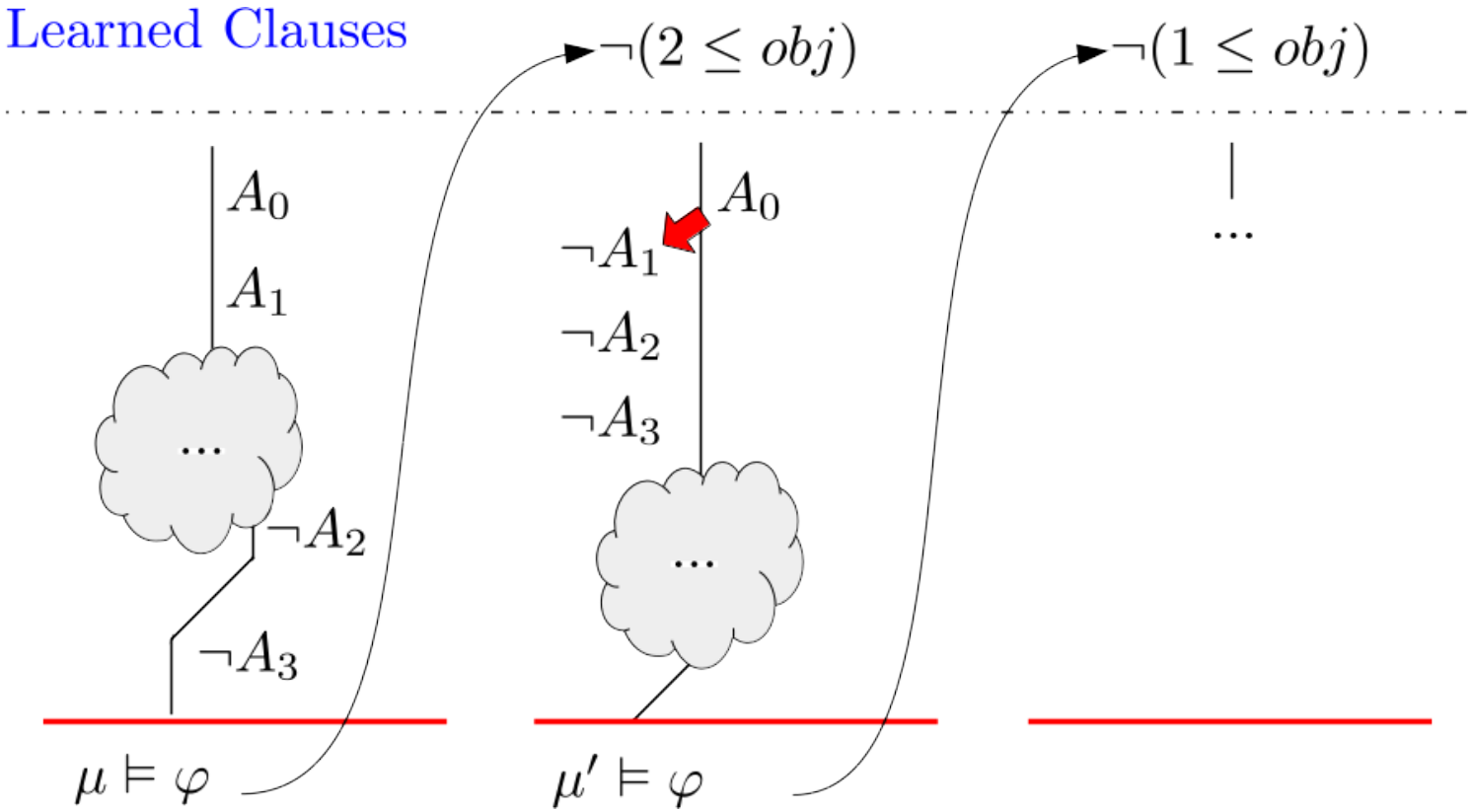
Example: with $k = 2$, $w = 1$ and $n = 4$



Running Example: OMT with sorting networks

- OPTIMIZATION STEP: learn $\neg(k \cdot w \leq obj)$ and restart/jump to *level* 0
- as soon as $k-1$ A_i are assigned $\top$
  $\Longrightarrow$ all others are unit-propagated to $\bot$

Example: with $k = 2$, $w = 1$ and $n = 4$



Solution: Combine OMT with Sorting Networks

Possible encodings for $n \times n$ Boolean relation $C(A_0, ..., A_{n-1}, B_0, ..., B_{n-1})$ are:

- **Bi-directional Sequential Counter** [21], in $O(n^2)$
  sum of A_i's, unary representation
- **Bi-directional Cardinality Network** [10, 3, 2], in $O(n\, log^2 n)$
  based on *merge-sort* algorithm idea

Solution: Combine OMT with Sorting Networks

Possible encodings for $n \times n$ Boolean relation $C(A_0, ..., A_{n-1}, B_0, ..., B_{n-1})$ are:

- **Bi-directional Sequential Counter** [21], in $O(n^2)$
  sum of A_i's, unary representation
- **Bi-directional Cardinality Network** [10, 3, 2], in $O(n \log^2 n)$
  based on *merge-sort* algorithm idea

Generalization

The same performance issue occurs for $\langle \varphi, obj \rangle$, where

$$obj = \tau_1 + ... + \tau_m,$$

$$\forall_j \in [1, m].\ (\tau_j = w_j \cdot \sum_{i=0}^{i=k_j} A_{ji}) \ \wedge \ (0 \leq \tau_j) \wedge (\tau_j \leq w_j \cdot k_j)$$

Solution:

- use a separate sorting circuit for each term τ_j
- add clauses in the form $(w_j \cdot i \leq \tau_j) \rightarrow (w_j \cdot i \leq obj)$

Contents

Experimental Evaluation

Test Framework:

- two *8-core 2.20Ghz Xeon* Linux machines with 64 GB of RAM

Tools:

- OPTIMATHSAT, OMT($\mathcal{LA}$) tool based on MATHSAT5 [8]
- νZ, general OMT solver based on Z3 [4]

(Partial) **Correctness Check**:

- all configurations agree on optimal lexicographic solution
  $\Longrightarrow$ ✓
- otherwise
  $\Longrightarrow$ used Z3 to check model is both *satisfiable* and *optimal*

Experiment #1

Benchmark-Set:

- Structured Learning Modulo Theories [22]: inference in hybrid domain

$$cover = \sum_i w_i A_i$$

$$obj = \sum_j w_j \cdot B_j + cover - \sum_k w_k \cdot C_k - |K - cover|$$

- 500 formulas, 600 *s*. timeout

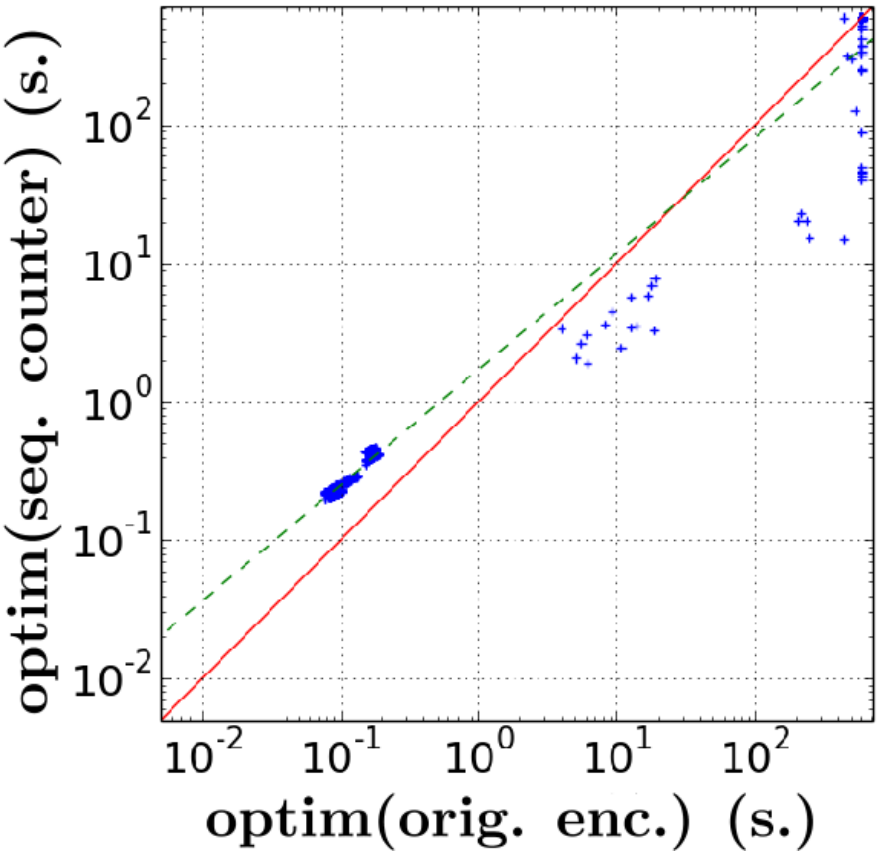
Experiment #1

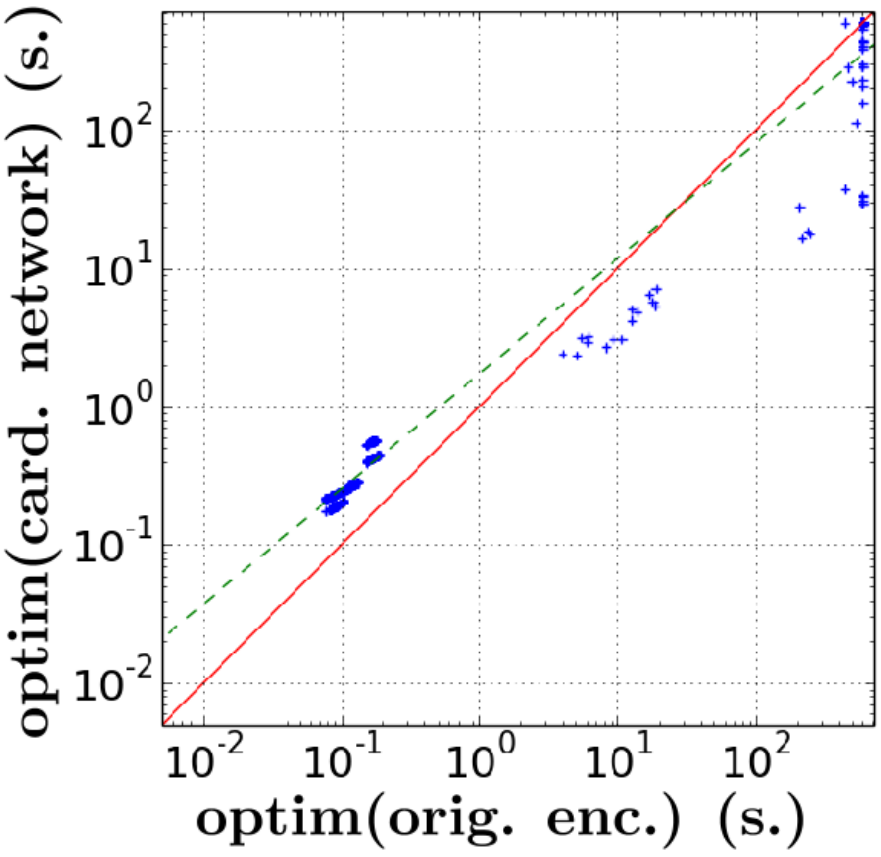
size	# total	# solved	# time-out	time (s.)
original encoding				
νZ	500	406	94	2120
OPTIMATHSAT	500	424	76	3522
OPTIMATHSAT using assert-soft				
orig. OMT enc.	500	421	79	2607
seq. counter enc.	500	441	59	6381
card. network enc.	500	**442**	58	6189

Observations

- use of *Sorting Networks*
  $\Longrightarrow$ improvement # solved benchmarks

Experiment #1





- use of *Sorting Networks*

  $\Longrightarrow$ more beneficial on harder benchmarks

Experiment #2

Benchmark-Set:

- Optimal Realization of a Goal Model enriched with Soft-Requirements [1]
  - Multiple Partial Weighted MaxSMT problems
  - 3-levels Lexicographic optimization
- 18996 generated formulas $\Longrightarrow$ 100 *s*. timeout

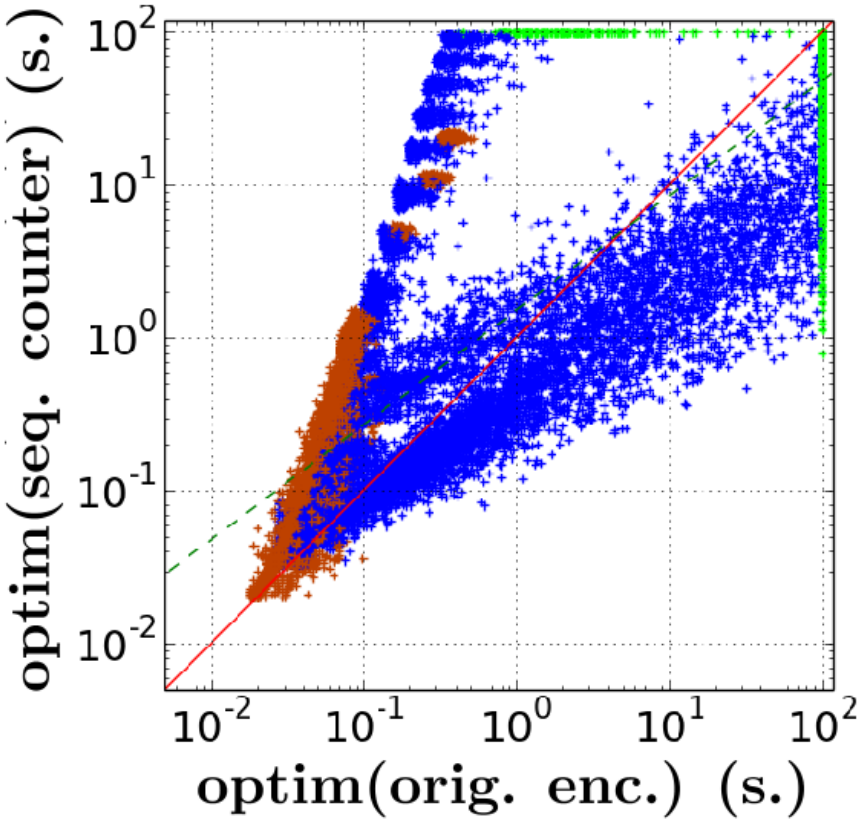
Experiment #2

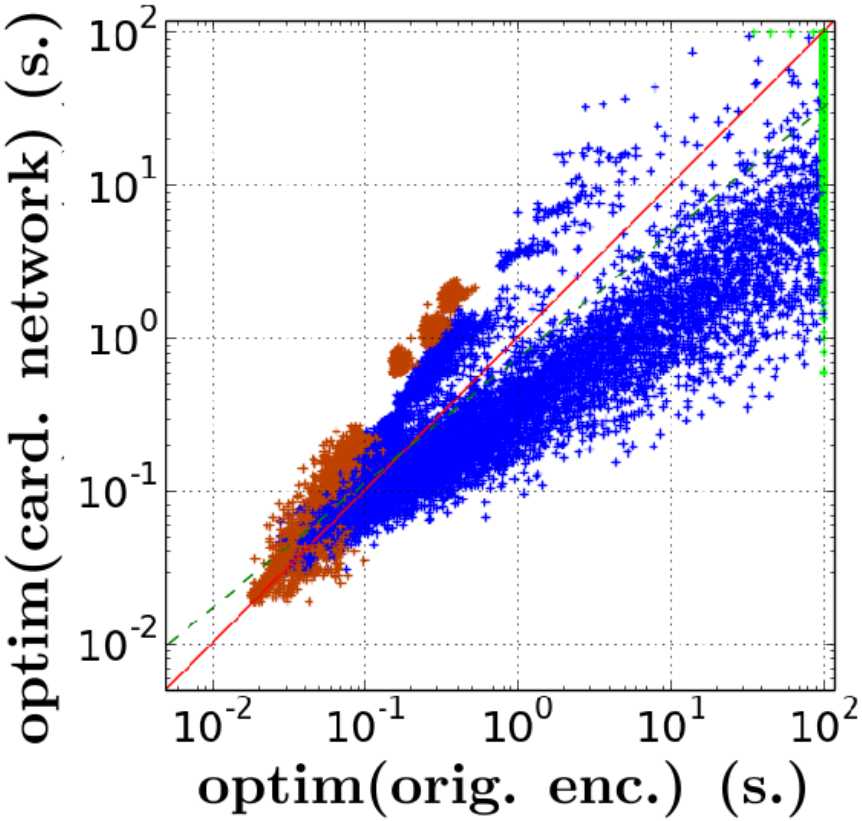
encoding	# inst.	# term.	# incorrect	time (s.)
OPTIMATHSAT				
orig. OMT enc.	18996	16316	0	48832
seq. counter enc.	18996	16929	0	90080
card. network enc.	18996	**17191**	0	39215
νZ				
maxres	18996	18996	255	1766
wmax	18996	16650	3785	38040

Observations

- use of *Sorting Networks*
  $\Longrightarrow$ larger # solved benchmarks
- OPTIMATHSAT + SN with *Cardinality Network* encoding
  $\Longrightarrow$ also faster than OPTIMATHSAT

Experiment #2





- *Sequential Counter* is expensive: $O(n^2)$

  solution $\Longrightarrow$ upper bound to circuit size

Experiment #2

circuit bound	# inst.	# term.	time (s.)
seq. counter enc.			
unbounded	18996	16929	90080
10 vars	18996	17033	39035
15 vars	18996	17061	39264
20 vars	18996	17152	43730
card. network enc.			
unbounded	18996	**17191**	39215
10 vars	18996	17058	36636
15 vars	18996	17133	37246
20 vars	18996	17190	39492

- *Sequential Counter*

  $\Longrightarrow$ improved speed & # solved benchmarks

- *Cardinality Network*

  $\Longrightarrow$ slightly worse

Conclusions & future wok

Conclusions:

OMT can benefit from *Sorting Networks* when dealing with

- Partial Weighted MaxSMT
- other Pseudo-Boolean objectives

Works also with SMT with PB objectives (no empirical data yet)

Future Work:

- extend this technique to better handle heterogeneous sets of weights values.

Thanks

Thanks for listening!

Questions..?

References I

[1] CGM-Tool.

www.cgm-tool.eu.

[2] I. Abío, R. Nieuwenhuis, A. Oliveras, and E. Rodríguez-Carbonell.

A Parametric Approach for Smaller and Better Encodings of Cardinality Constraints.

In *19th International Conference on Principles and Practice of Constraint Programming*, CP'13, 2013.

[3] R. Asín, R. Nieuwenhuis, A. Oliveras, and E. Rodríguez-Carbonell.

Cardinality networks: a theoretical and empirical study.

Constraints, 16(2):195–221, 2011.

[4] N. Bjorner and A.-D. Phan.

νZ - Maximal Satisfaction with Z3.

In *Proc International Symposium on Symbolic Computation in Software Science*, Gammart, Tunisia, December 2014. EasyChair Proceedings in Computing (EPiC).

http://www.easychair.org/publications/?page=862275542.

References II

[5] N. Bjorner, A.-D. Phan, and L. Fleckenstein.

νZ - An Optimizing SMT Solver.

In *Proc. TACAS*, volume 9035 of *LNCS*. Springer, 2015.

[6] A. Cimatti, A. Franzén, A. Griggio, R. Sebastiani, and C. Stenico.

Satisfiability modulo the theory of costs: Foundations and applications.

In *TACAS*, volume 6015 of *LNCS*, pages 99–113. Springer, 2010.

[7] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani.

A Modular Approach to MaxSAT Modulo Theories.

In *International Conference on Theory and Applications of Satisfiability Testing, SAT*, volume 7962 of *LNCS*, July 2013.

[8] A. Cimatti, A. Griggio, B. J. Schaafsma, and R. Sebastiani.

The MathSAT 5 SMT Solver.

In *Tools and Algorithms for the Construction and Analysis of Systems, TACAS'13.*, volume 7795 of *LNCS*, pages 95–109. Springer, 2013.

References III

[9] J. Corander, T. Janhunen, J. Rintanen, H. J. Nyman, and J. Pensar.

Learning chordal markov networks by constraint satisfaction.

CoRR, abs/1310.0927, 2013.

[10] N. Eén and N. Sörensson.

Translating pseudo-boolean constraints into SAT.

JSAT, 2(1-4):1–26, 2006.

[11] J. Henry, M. Asavoae, D. Monniaux, and C. Maïza.

How to compute worst-case execution time by optimization modulo theory and a clever encoding of program semantics.

SIGPLAN Not., 49(5):43–52, June 2014.

[12] A. S. Köksal, V. Kuncak, and P. Suter.

Constraints as control.

In *POPL*, pages 151–164, 2012.

References IV

[13] Y. Li, A. Albarghouthi, Z. Kincad, A. Gurfinkel, and M. Chechik.

Symbolic Optimization with SMT Solvers.

In *POPL*, 2014.

[14] Y. Li, A. Albarghouthi, Z. Kincaid, A. Gurfinkel, and M. Chechik.

Symbolic optimization with smt solvers.

In *POPL*, pages 607–618, 2014.

[15] R. Nieuwenhuis and A. Oliveras.

On SAT Modulo Theories and Optimization Problems.

In *Proc. Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *LNCS*. Springer, 2006.

[16] R. Sebastiani and S. Tomasi.

Optimization in SMT with LA(Q) Cost Functions.

In *IJCAR*, volume 7364 of *LNAI*, pages 484–498. Springer, July 2012.

References V

[17] R. Sebastiani and S. Tomasi.

Optimization Modulo Theories with Linear Rational Costs.

ACM Transactions on Computational Logics, 16(2), March 2015.

[18] R. Sebastiani and S. Tomasi.

Optimization Modulo Theories with Linear Rational Costs.

ACM Transactions on Computational Logics, 16(2), March 2015.

[19] R. Sebastiani and P. Trentin.

OptiMathSAT: A Tool for Optimization Modulo Theories.

In *Proc. International Conference on Computer-Aided Verification, CAV 2015*, volume 9206 of *LNCS*. Springer, 2015.

[20] R. Sebastiani and P. Trentin.

Pushing the Envelope of Optimization Modulo Theories with Linear-Arithmetic Cost Functions.

In *Proc. Int. Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'15*, volume 9035 of *LNCS*. Springer, 2015.

References VI

[21] C. Sinz.

Towards an optimal cnf encoding of boolean cardinality constraints.

In P. van Beek, editor, *CP*, volume 3709 of *Lecture Notes in Computer Science*, pages 827–831. Springer, 2005.

[22] S. Teso, R. Sebastiani, and A. Passerini.

Structured Learning Modulo Theories.

Artificial Intelligence Journal, 2015.

To appear.