Computing with Oracles

From NP to Beyond NP and Back Again

Joao Marques-Silva

University of Lisbon, Portugal

Beyond NP Meeting

Paris, France

10 May 2017

The SAT disruption

- SAT is NP-complete [Cook'71]

The SAT disruption

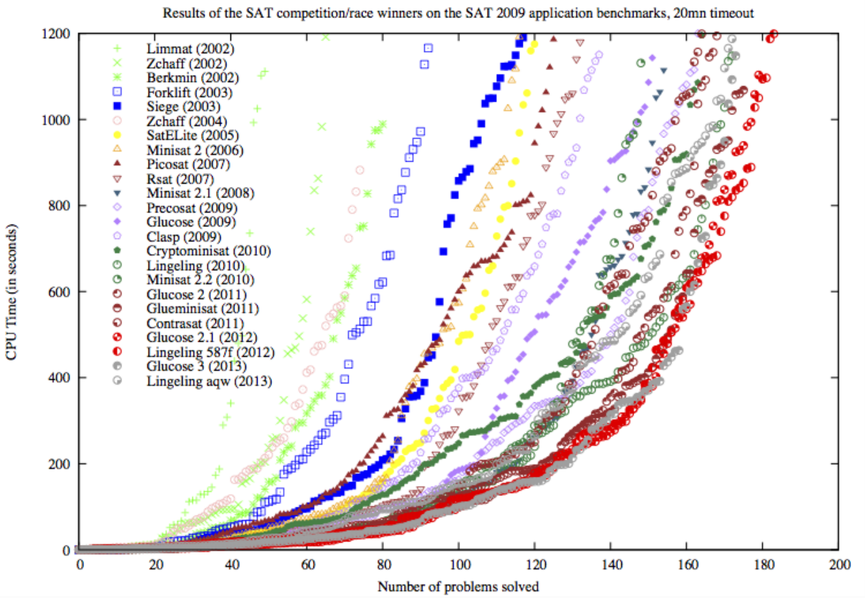
- SAT is NP-complete [Cook'71]
  - But, CDCL SAT solving is a success story of Computer Science

The SAT disruption

- SAT is NP-complete [Cook'71]
  - But, CDCL SAT solving is a success story of Computer Science
  - Hundreds (thousands?) of practical applications

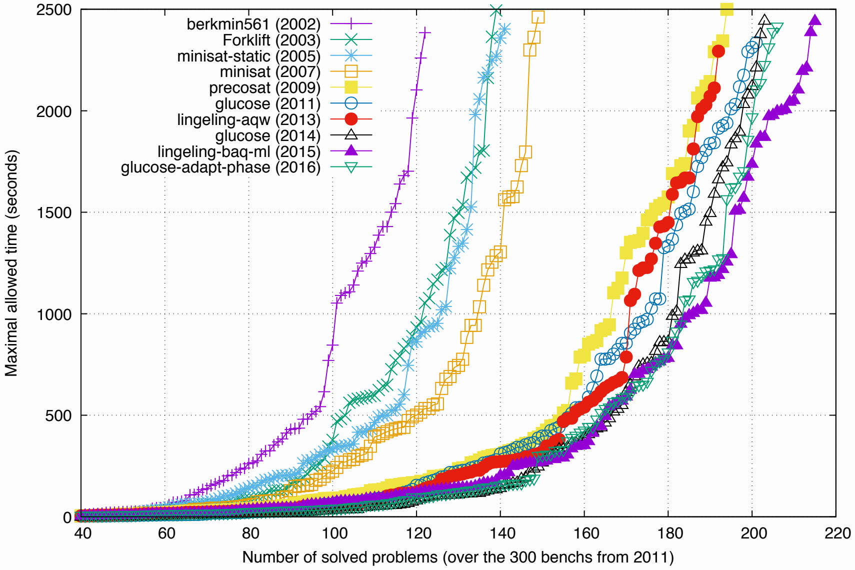
SAT solver improvement I

[Source: Le Berre 2013]

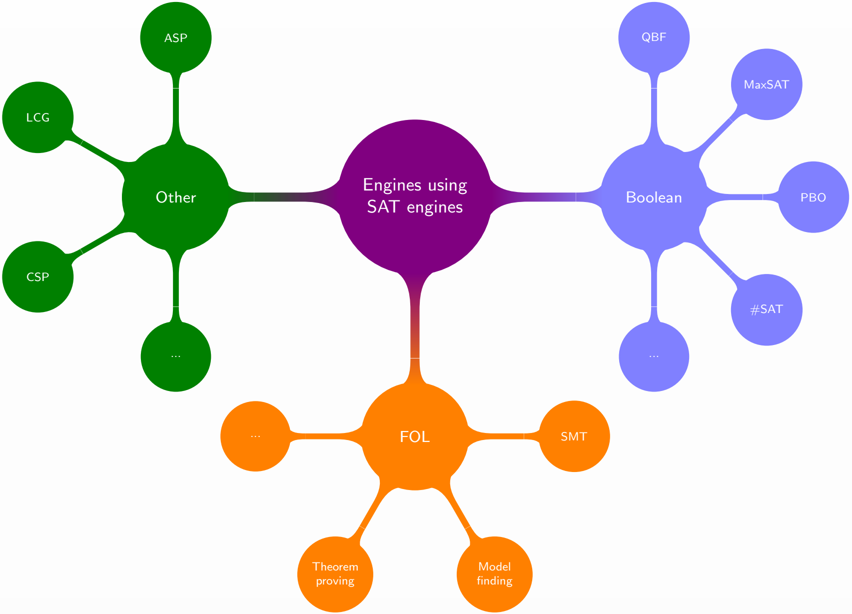


SAT solver improvement II

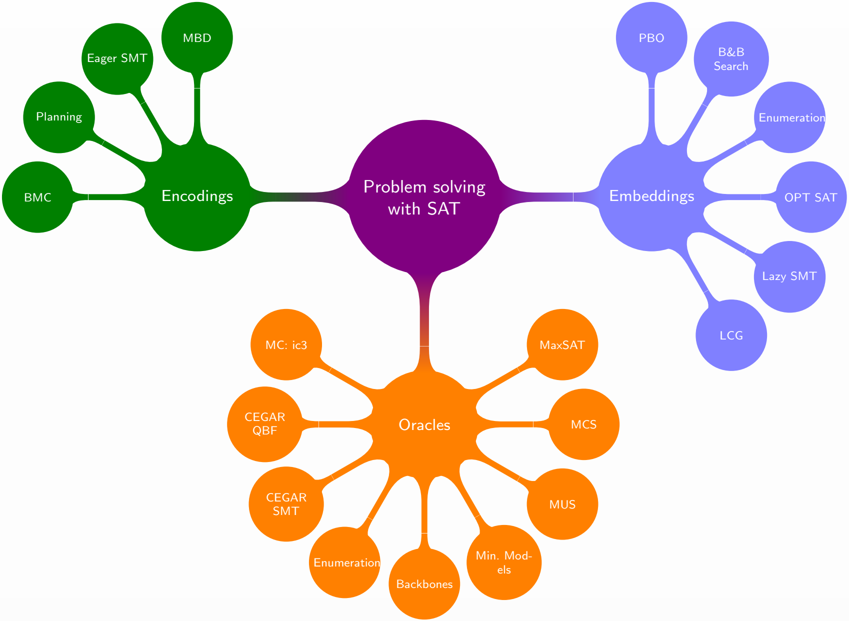
[Source: Simon 2015]



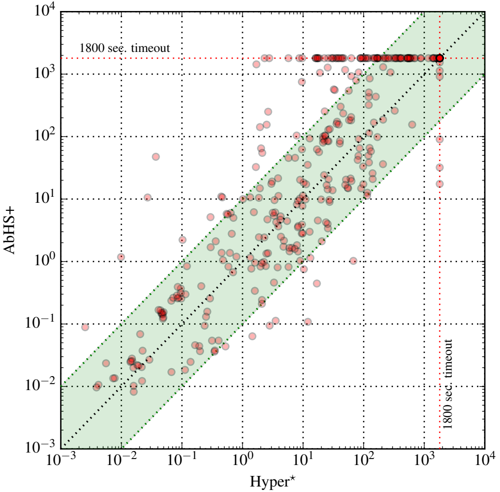
SAT is **the** engines' engine



SAT is ubiquitous in problem solving

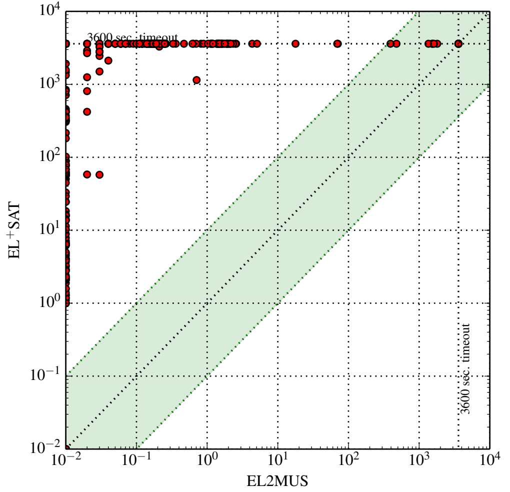


SAT can make the difference – propositional abduction



- Topic(s): quantified optimization [ECAI'16]
- Instances: KR16 propositional abduction

SAT can make the difference – axiom pinpointing



- Topic(s): MUS enumeration; MCSes; implicit hitting sets [SAT'15]
- Instances: $\mathcal{EL}^+$ medical ontologies

Part I

From NP to Beyond NP

Outline – part A

Background

MaxSAT Solving

2QBF Solving

Outline

Background

MaxSAT Solving

2QBF Solving

Beyond decision problems

Answer	Problem Type

Beyond decision problems

Answer	Problem Type
Yes/No	Decision Problems

Beyond decision problems

Answer	Problem Type
Yes/No	Decision Problems
Some solution	

Beyond decision problems

Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems

Beyond decision problems

Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems
All solutions	

Beyond decision problems

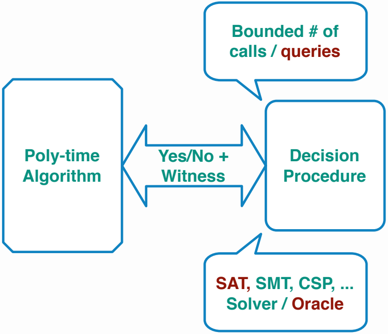
Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems
All solutions	Enumeration Problems

… and beyond NP – decision and function problems

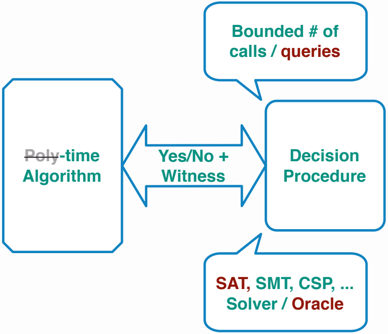
$$
\begin{tikzcd}
 & \vdots & \\
\Sigma_3^p & & \Pi_3^p \\
 & \Delta_3^p \arrow[ul] \arrow[ur] & \\
\Sigma_2^p \arrow[uu] \arrow[ur] & & \Pi_2^p \arrow[uu] \arrow[ul] \\
 & P^{NP} = \Delta_2^p \arrow[ul] \arrow[ur] & \\
NP = \Sigma_1^p \arrow[uu] \arrow[ur] & & \Pi_1^p = coNP \arrow[uu] \arrow[ul] \\
 & \Delta_0^p = \Sigma_0^p = P = \Pi_0^p = \Delta_1^p \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

$$
\begin{tikzcd}
 & \vdots & \\
F\Sigma_3^p & & F\Pi_3^p \\
 & F\Delta_3^p \arrow[ul] \arrow[ur] & \\
F\Sigma_2^p \arrow[uu] \arrow[ur] & & F\Pi_2^p \arrow[uu] \arrow[ul] \\
 & FP^{NP} = F\Delta_2^p \arrow[ul] \arrow[ur] & \\
FNP = F\Sigma_1^p \arrow[uu] \arrow[ur] & & F\Pi_1^p = coFNP \arrow[uu] \arrow[ul] \\
 & F\Delta_0^p = F\Sigma_0^p = FP = F\Pi_0^p = F\Delta_1^p \arrow[ul] \arrow[ur] &
\end{tikzcd}
$$

Oracle-based problem solving – ideal scenario



Oracle-based problem solving – in some settings



Many problems to solve – within FP^{NP}

Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems
All solutions	Enumeration Problems

Many problems to solve – within $\mathrm{FP}^{\mathrm{NP}}$

Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems
All solutions	Enumeration Problems

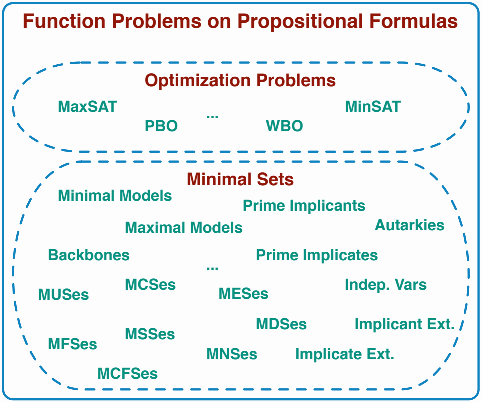
Function Problems on Propositional Formulas

MaxSAT PBO ... WBO MinSAT

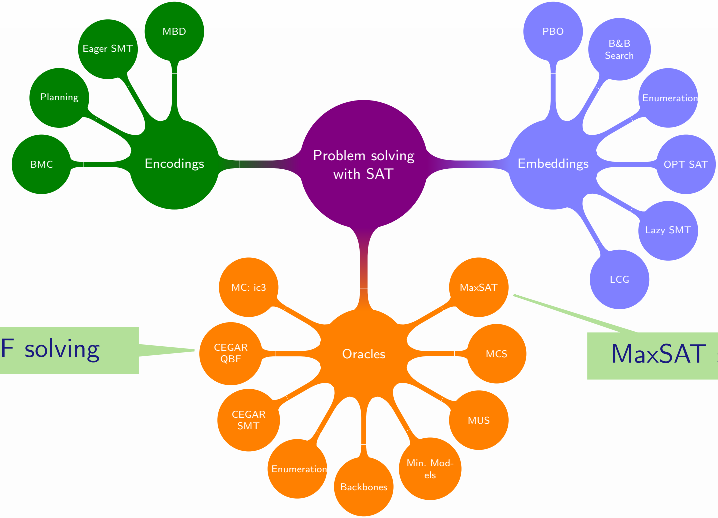
Minimal Models Prime Implicants Maximal Models Autarkies Backbones ... Prime Implicates MUSes MCSes MESes Indep. Vars MDSes Implicant Ext. MFSes MSSes MNSes Implicate Ext. MCFSes

Many problems to solve – within FP^{NP}

Answer	Problem Type
Yes/No	Decision Problems
Some solution	Function Problems
All solutions	Enumeration Problems



Selection of topics



2QBF solving

MaxSAT solving

Outline

Background

MaxSAT Solving

2QBF Solving

Recap MaxSAT

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Given unsatisfiable formula, find largest subset of clauses that is satisfiable

Recap MaxSAT

$\boxed{x_6 \vee x_2}$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\boxed{\neg x_4 \vee x_5}$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Given unsatisfiable formula, find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula

Recap MaxSAT

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\boxed{\neg x_1}$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$\boxed{x_7 \vee x_5}$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Given unsatisfiable formula, find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest MCSes

Recap MaxSAT

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

- Given unsatisfiable formula, find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest MCSes
  - **Note**: Clauses can have weights & there can be hard clauses

Recap MaxSAT

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

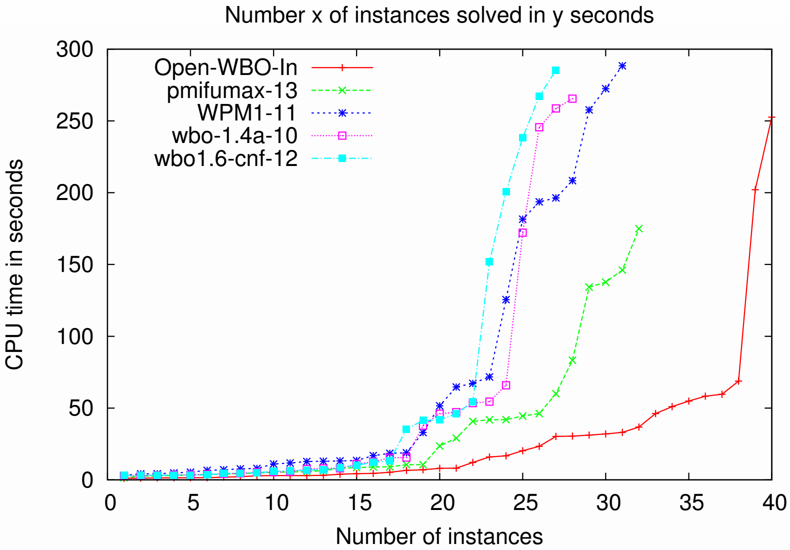
- Given unsatisfiable formula, find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest **cost** MCSes
  - **Note**: Clauses can have weights & there can be hard clauses

Recap MaxSAT

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

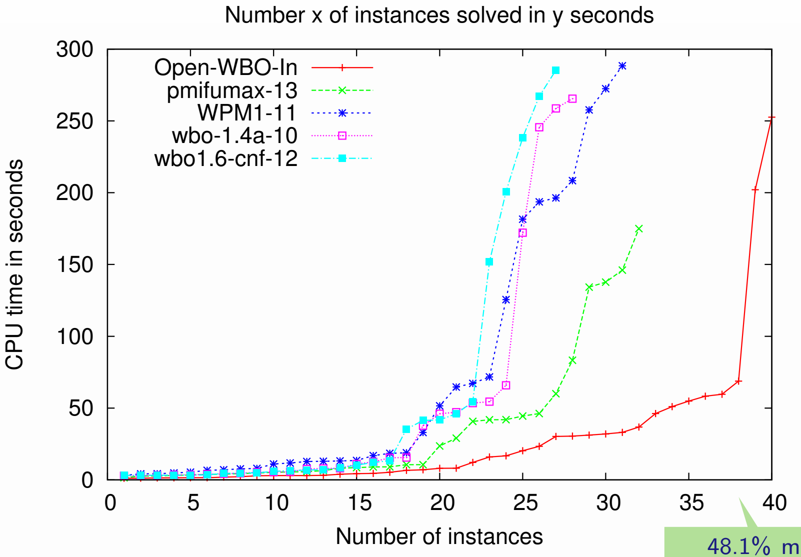
- Given unsatisfiable formula, find largest subset of clauses that is satisfiable
- A Minimal Correction Subset (**MCS**) is an irreducible relaxation of the formula
- The MaxSAT solution is one of the smallest **cost** MCSes
  - **Note**: Clauses can have weights & there can be hard clauses
- **Many** practical applications

The MaxSAT (r)evolution – plain industrial instances



Source: [MaxSAT 2014 organizers]

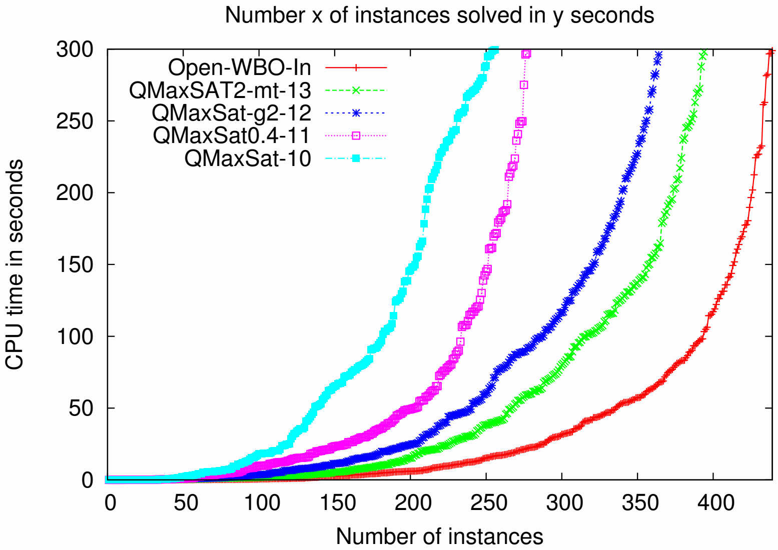
The MaxSAT (r)evolution – plain industrial instances



Source: [MaxSAT 2014 organizers]

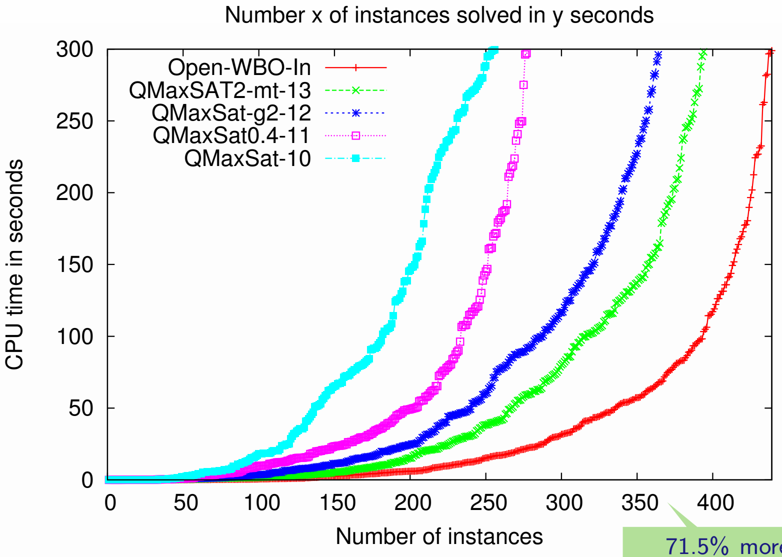
48.1% more instances solved!

The MaxSAT (r)evolution – partial



Source: [MaxSAT 2014 organizers]

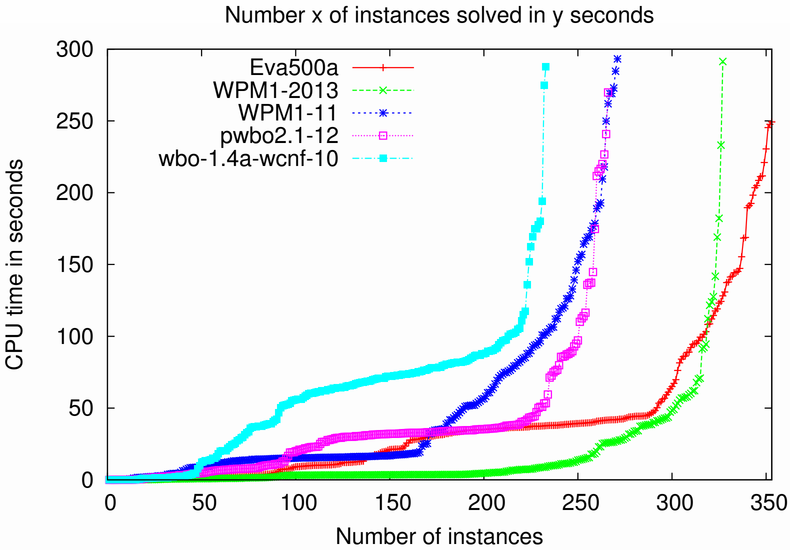
The MaxSAT (r)evolution – partial



Source: [MaxSAT 2014 organizers]

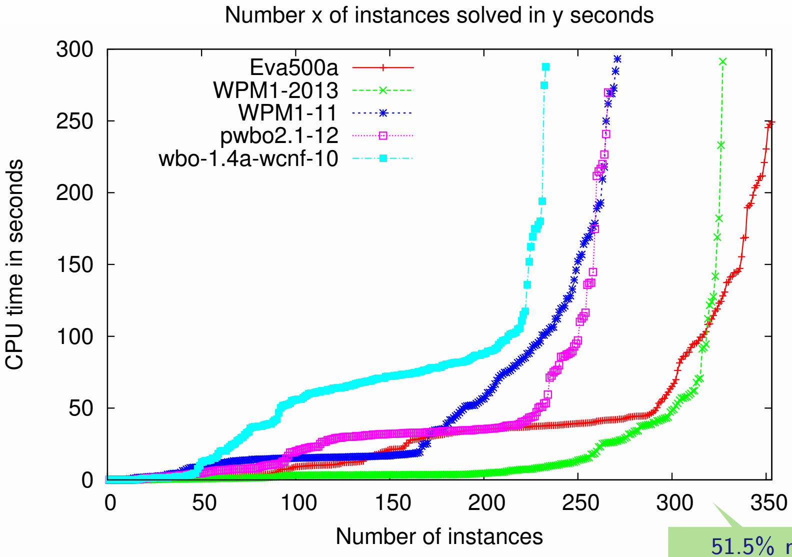
71.5% more instances solved!

The MaxSAT (r)evolution – weighted partial



Source: [MaxSAT 2014 organizers]

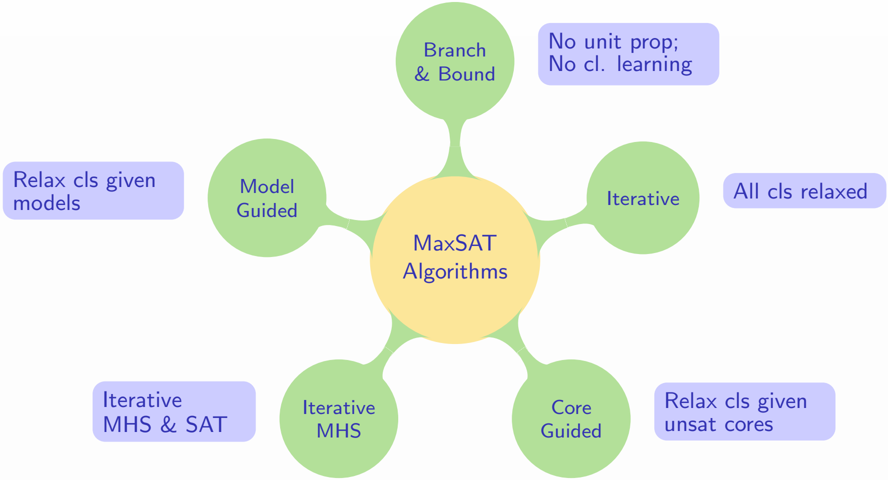
The MaxSAT (r)evolution – weighted partial



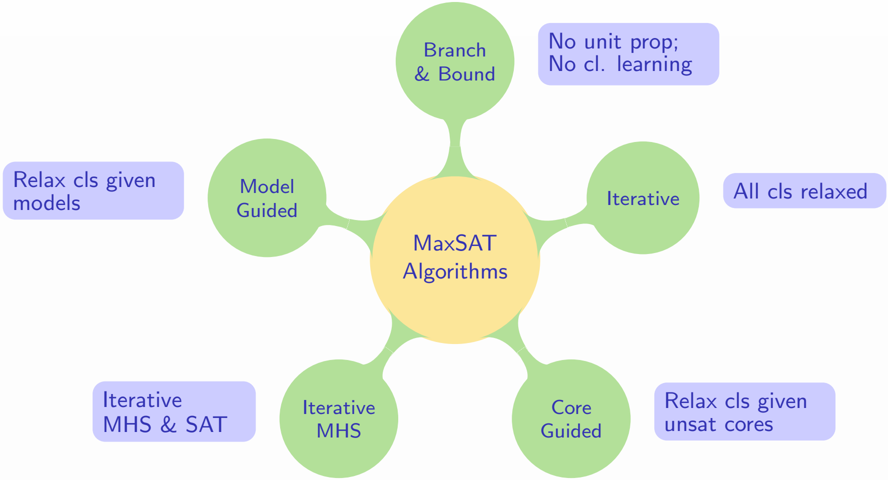
Source: [MaxSAT 2014 organizers]

51.5% more instances solved!

Many MaxSAT approaches



Many MaxSAT approaches



- For practical (**industrial**) instances: **core-guided** approaches are the most effective [MaxSAT14]

Outline

Background

MaxSAT Solving
- Iterative SAT Solving
- Core-Guided Algorithms
- Minimum Hitting Sets

2QBF Solving

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

Example CNF formula

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$	$\neg x_6 \vee x_2 \vee r_2$	$\neg x_2 \vee x_1 \vee r_3$	$\neg x_1 \vee r_4$
$\neg x_6 \vee x_8 \vee r_5$	$x_6 \vee \neg x_8 \vee r_6$	$x_2 \vee x_4 \vee r_7$	$\neg x_4 \vee x_5 \vee r_8$
$x_7 \vee x_5 \vee r_9$	$\neg x_7 \vee x_5 \vee r_{10}$	$\neg x_5 \vee x_3 \vee r_{11}$	$\neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 12$

Relax all clauses; Set $UB = 12 + 1$

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$	$\neg x_6 \vee x_2 \vee r_2$	$\neg x_2 \vee x_1 \vee r_3$	$\neg x_1 \vee r_4$
$\neg x_6 \vee x_8 \vee r_5$	$x_6 \vee \neg x_8 \vee r_6$	$x_2 \vee x_4 \vee r_7$	$\neg x_4 \vee x_5 \vee r_8$
$x_7 \vee x_5 \vee r_9$	$\neg x_7 \vee x_5 \vee r_{10}$	$\neg x_5 \vee x_3 \vee r_{11}$	$\neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 12$

Formula is SAT; E.g. all $x_i = 0$ and $r_1 = r_7 = r_9 = 1$ (i.e. cost $= 3$)

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$	$\neg x_6 \vee x_2 \vee r_2$	$\neg x_2 \vee x_1 \vee r_3$	$\neg x_1 \vee r_4$
$\neg x_6 \vee x_8 \vee r_5$	$x_6 \vee \neg x_8 \vee r_6$	$x_2 \vee x_4 \vee r_7$	$\neg x_4 \vee x_5 \vee r_8$
$x_7 \vee x_5 \vee r_9$	$\neg x_7 \vee x_5 \vee r_{10}$	$\neg x_5 \vee x_3 \vee r_{11}$	$\neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 2$

Refine $UB = 3$

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$	$\neg x_6 \vee x_2 \vee r_2$	$\neg x_2 \vee x_1 \vee r_3$	$\neg x_1 \vee r_4$
$\neg x_6 \vee x_8 \vee r_5$	$x_6 \vee \neg x_8 \vee r_6$	$x_2 \vee x_4 \vee r_7$	$\neg x_4 \vee x_5 \vee r_8$
$x_7 \vee x_5 \vee r_9$	$\neg x_7 \vee x_5 \vee r_{10}$	$\neg x_5 \vee x_3 \vee r_{11}$	$\neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 2$

Formula is SAT; E.g. $x_1 = x_2 = 1$; $x_3 = ... = x_8 = 0$ and $r_4 = r_9 = 1$ (i.e. cost $= 2$)

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$ $\quad \neg x_6 \vee x_2 \vee r_2$ $\quad \neg x_2 \vee x_1 \vee r_3$ $\quad \neg x_1 \vee r_4$

$\neg x_6 \vee x_8 \vee r_5$ $\quad x_6 \vee \neg x_8 \vee r_6$ $\quad x_2 \vee x_4 \vee r_7$ $\quad \neg x_4 \vee x_5 \vee r_8$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_{11}$ $\quad \neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 1$

Refine $UB = 2$

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$ $\quad \neg x_6 \vee x_2 \vee r_2$ $\quad \neg x_2 \vee x_1 \vee r_3$ $\quad \neg x_1 \vee r_4$

$\neg x_6 \vee x_8 \vee r_5$ $\quad x_6 \vee \neg x_8 \vee r_6$ $\quad x_2 \vee x_4 \vee r_7$ $\quad \neg x_4 \vee x_5 \vee r_8$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_{11}$ $\quad \neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 1$

Formula is UNSAT; terminate

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$ $\quad \neg x_6 \vee x_2 \vee r_2$ $\quad \neg x_2 \vee x_1 \vee r_3$ $\quad \neg x_1 \vee r_4$

$\neg x_6 \vee x_8 \vee r_5$ $\quad x_6 \vee \neg x_8 \vee r_6$ $\quad x_2 \vee x_4 \vee r_7$ $\quad \neg x_4 \vee x_5 \vee r_8$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_{11}$ $\quad \neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 1$

MaxSAT solution is last satisfied UB: $UB = 2$

Basic MaxSAT with iterative SAT solving

$x_6 \vee x_2 \vee r_1$	$\neg x_6 \vee x_2 \vee r_2$	$\neg x_2 \vee x_1 \vee r_3$	$\neg x_1 \vee r_4$
$\neg x_6 \vee x_8 \vee r_5$	$x_6 \vee \neg x_8 \vee r_6$	$x_2 \vee x_4 \vee r_7$	$\neg x_4 \vee x_5 \vee r_8$
$x_7 \vee x_5 \vee r_9$	$\neg x_7 \vee x_5 \vee r_{10}$	$\neg x_5 \vee x_3 \vee r_{11}$	$\neg x_3 \vee r_{12}$

$\sum_{i=1}^{12} r_i \leq 1$

MaxSAT solution is last satisfied UB: $UB = 2$

AtMost$k$/PB constraints over **all** relaxation variables

All (possibly many) soft clauses relaxed

Outline

Background

MaxSAT Solving

- Iterative SAT Solving
- Core-Guided Algorithms
- Minimum Hitting Sets

2QBF Solving

MSU3 core-guided algorithm

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

Example CNF formula

MSU3 core-guided algorithm

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1$	$\neg x_1$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4$	$\neg x_4 \vee x_5$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3$	$\neg x_3$

Formula is UNSAT; $\text{OPT} \leq |\varphi| - 1$; Get unsat core

MSU3 core-guided algorithm

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1 \vee r_1$	$\neg x_1 \vee r_2$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4 \vee r_3$	$\neg x_4 \vee x_5 \vee r_4$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3 \vee r_5$	$\neg x_3 \vee r_6$

$\sum_{i=1}^{6} r_i \leq 1$

Add relaxation variables and AtMost$k$, $k = 1$, constraint

MSU3 core-guided algorithm

$x_6 \vee x_2$	$\neg x_6 \vee x_2$	$\neg x_2 \vee x_1 \vee r_1$	$\neg x_1 \vee r_2$
$\neg x_6 \vee x_8$	$x_6 \vee \neg x_8$	$x_2 \vee x_4 \vee r_3$	$\neg x_4 \vee x_5 \vee r_4$
$x_7 \vee x_5$	$\neg x_7 \vee x_5$	$\neg x_5 \vee x_3 \vee r_5$	$\neg x_3 \vee r_6$

$\sum_{i=1}^{6} r_i \leq 1$

Formula is (again) UNSAT; $\text{OPT} \leq |\varphi| - 2$; Get unsat core

MSU3 core-guided algorithm

$x_6 \vee x_2 \vee r_7$ $\quad \neg x_6 \vee x_2 \vee r_8$ $\quad \neg x_2 \vee x_1 \vee r_1$ $\quad \neg x_1 \vee r_2$

$\neg x_6 \vee x_8$ $\quad x_6 \vee \neg x_8$ $\quad x_2 \vee x_4 \vee r_3$ $\quad \neg x_4 \vee x_5 \vee r_4$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_5$ $\quad \neg x_3 \vee r_6$

$\sum_{i=1}^{10} r_i \leq 2$

Add new relaxation variables and update AtMost$k$, k=2, constraint

MSU3 core-guided algorithm

$x_6 \vee x_2 \vee r_7$ $\quad$ $\neg x_6 \vee x_2 \vee r_8$ $\quad$ $\neg x_2 \vee x_1 \vee r_1$ $\quad$ $\neg x_1 \vee r_2$

$\neg x_6 \vee x_8$ $\quad$ $x_6 \vee \neg x_8$ $\quad$ $x_2 \vee x_4 \vee r_3$ $\quad$ $\neg x_4 \vee x_5 \vee r_4$

$x_7 \vee x_5 \vee r_9$ $\quad$ $\neg x_7 \vee x_5 \vee r_{10}$ $\quad$ $\neg x_5 \vee x_3 \vee r_5$ $\quad$ $\neg x_3 \vee r_6$

$\sum_{i=1}^{10} r_i \leq 2$

Instance is now SAT

MSU3 core-guided algorithm

$x_6 \vee x_2 \vee r_7$ $\quad \neg x_6 \vee x_2 \vee r_8$ $\quad \neg x_2 \vee x_1 \vee r_1$ $\quad \neg x_1 \vee r_2$

$\neg x_6 \vee x_8$ $\quad x_6 \vee \neg x_8$ $\quad x_2 \vee x_4 \vee r_3$ $\quad \neg x_4 \vee x_5 \vee r_4$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_5$ $\quad \neg x_3 \vee r_6$

$\sum_{i=1}^{10} r_i \leq 2$

MaxSAT solution is $|\varphi| - \mathcal{I} = 12 - 2 = 10$

MSU3 core-guided algorithm

$x_6 \vee x_2 \vee r_7$ $\quad \neg x_6 \vee x_2 \vee r_8$ $\quad \neg x_2 \vee x_1 \vee r_1$ $\quad \neg x_1 \vee r_2$

$\neg x_6 \vee x_8$ $\quad x_6 \vee \neg x_8$ $\quad x_2 \vee x_4 \vee r_3$ $\quad \neg x_4 \vee x_5 \vee r_4$

$x_7 \vee x_5 \vee r_9$ $\quad \neg x_7 \vee x_5 \vee r_{10}$ $\quad \neg x_5 \vee x_3 \vee r_5$ $\quad \neg x_3 \vee r_6$

$\sum_{i=1}^{10} r_i \leq 2$

MaxSAT solution is $|\varphi| - \mathcal{I} = 12 - 2 = 10$

AtMost$k$/PB constraints used

Relaxed soft clauses become **hard**

MSU3 core-guided algorithm

$x_6 \vee x_2 \vee r_7$ $\quad$ $\neg x_6 \vee x_2 \vee r_8$ $\quad$ $\neg x_2 \vee x_1 \vee r_1$ $\quad$ $\neg x_1 \vee r_2$

$\boxed{\neg x_6 \vee x_8 \quad x_6 \vee \neg x_8}$ $\quad$ $x_2 \vee x_4 \vee r_3$ $\quad$ $\neg x_4 \vee x_5 \vee r_4$

$x_7 \vee x_5 \vee r_9$ $\quad$ $\neg x_7 \vee x_5 \vee r_{10}$ $\quad$ $\neg x_5 \vee x_3 \vee r_5$ $\quad$ $\neg x_3 \vee r_6$

$\sum_{i=1}^{10} r_i \leq 2$

MaxSAT solution is $|\varphi| - \mathcal{I} = 12 - 2 = 10$

AtMost$k$/PB constraints used

Some clauses not relaxed

Relaxed soft clauses become **hard**

Outline

Background

MaxSAT Solving

- Iterative SAT Solving
- Core-Guided Algorithms
- Minimum Hitting Sets

2QBF Solving

MHS approach for MaxSAT

$$
\begin{array}{llll}
c_1 = x_6 \vee x_2 & c_2 = \neg x_6 \vee x_2 & c_3 = \neg x_2 \vee x_1 & c_4 = \neg x_1 \\
c_5 = \neg x_6 \vee x_8 & c_6 = x_6 \vee \neg x_8 & c_7 = x_2 \vee x_4 & c_8 = \neg x_4 \vee x_5 \\
c_9 = x_7 \vee x_5 & c_{10} = \neg x_7 \vee x_5 & c_{11} = \neg x_5 \vee x_3 & c_{12} = \neg x_3
\end{array}
$$

$$\mathcal{K} = \emptyset$$

- Find MHS of $\mathcal{K}$:

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \emptyset$$

- Find MHS of $\mathcal{K}$: $\emptyset$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \emptyset$$

- Find MHS of $\mathcal{K}$: $\emptyset$
- SAT$(\mathcal{F} \setminus \emptyset)$?

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \emptyset$$

- Find MHS of $\mathcal{K}$: $\emptyset$
- SAT($\mathcal{F} \setminus \emptyset$)? No

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \emptyset$$

- Find MHS of $\mathcal{K}$: $\emptyset$
- SAT($\mathcal{F} \setminus \emptyset$)? No
- Core of $\mathcal{F}$: $\{c_1, c_2, c_3, c_4\}$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$: $\emptyset$
- SAT($\mathcal{F} \setminus \emptyset$)? No
- Core of $\mathcal{F}$: $\{c_1, c_2, c_3, c_4\}$. Update $\mathcal{K}$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$ $\quad$ $c_2 = \neg x_6 \vee x_2$ $\quad$ $c_3 = \neg x_2 \vee x_1$ $\quad$ $c_4 = \neg x_1$

$c_5 = \neg x_6 \vee x_8$ $\quad$ $c_6 = x_6 \vee \neg x_8$ $\quad$ $c_7 = x_2 \vee x_4$ $\quad$ $c_8 = \neg x_4 \vee x_5$

$c_9 = x_7 \vee x_5$ $\quad$ $c_{10} = \neg x_7 \vee x_5$ $\quad$ $c_{11} = \neg x_5 \vee x_3$ $\quad$ $c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$:

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1\}$

MHS approach for MaxSAT

$$c_1 = x_6 \lor x_2 \qquad c_2 = \neg x_6 \lor x_2 \qquad c_3 = \neg x_2 \lor x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \lor x_8 \qquad c_6 = x_6 \lor \neg x_8 \qquad c_7 = x_2 \lor x_4 \qquad c_8 = \neg x_4 \lor x_5$$

$$c_9 = x_7 \lor x_5 \qquad c_{10} = \neg x_7 \lor x_5 \qquad c_{11} = \neg x_5 \lor x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1\}$
- SAT($\mathcal{F} \setminus \{c_1\}$)?

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1\}$
- SAT($\mathcal{F} \setminus \{c_1\}$)? No

MHS approach for MaxSAT

$$
\begin{array}{llll}
c_1 = x_6 \vee x_2 & c_2 = \neg x_6 \vee x_2 & c_3 = \neg x_2 \vee x_1 & c_4 = \neg x_1 \\
c_5 = \neg x_6 \vee x_8 & c_6 = x_6 \vee \neg x_8 & c_7 = x_2 \vee x_4 & c_8 = \neg x_4 \vee x_5 \\
c_9 = x_7 \vee x_5 & c_{10} = \neg x_7 \vee x_5 & c_{11} = \neg x_5 \vee x_3 & c_{12} = \neg x_3
\end{array}
$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1\}$
- SAT($\mathcal{F} \setminus \{c_1\}$)? No
- Core of $\mathcal{F}$: $\{c_9, c_{10}, c_{11}, c_{12}\}$

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1\}$
- SAT($\mathcal{F} \setminus \{c_1\}$)? No
- Core of $\mathcal{F}$: $\{c_9, c_{10}, c_{11}, c_{12}\}$. Update $\mathcal{K}$

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$:

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$ $\quad$ $c_2 = \neg x_6 \vee x_2$ $\quad$ $c_3 = \neg x_2 \vee x_1$ $\quad$ $c_4 = \neg x_1$

$c_5 = \neg x_6 \vee x_8$ $\quad$ $c_6 = x_6 \vee \neg x_8$ $\quad$ $c_7 = x_2 \vee x_4$ $\quad$ $c_8 = \neg x_4 \vee x_5$

$c_9 = x_7 \vee x_5$ $\quad$ $c_{10} = \neg x_7 \vee x_5$ $\quad$ $c_{11} = \neg x_5 \vee x_3$ $\quad$ $c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1, c_9\}$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$ $\quad c_2 = \neg x_6 \vee x_2$ $\quad c_3 = \neg x_2 \vee x_1$ $\quad c_4 = \neg x_1$

$c_5 = \neg x_6 \vee x_8$ $\quad c_6 = x_6 \vee \neg x_8$ $\quad c_7 = x_2 \vee x_4$ $\quad c_8 = \neg x_4 \vee x_5$

$c_9 = x_7 \vee x_5$ $\quad c_{10} = \neg x_7 \vee x_5$ $\quad c_{11} = \neg x_5 \vee x_3$ $\quad c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1, c_9\}$
- SAT($\mathcal{F} \setminus \{c_1, c_9\}$)?

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1, c_9\}$
- SAT($\mathcal{F} \setminus \{c_1, c_9\}$)? No

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$ $\quad c_2 = \neg x_6 \vee x_2$ $\quad c_3 = \neg x_2 \vee x_1$ $\quad c_4 = \neg x_1$

$c_5 = \neg x_6 \vee x_8$ $\quad c_6 = x_6 \vee \neg x_8$ $\quad c_7 = x_2 \vee x_4$ $\quad c_8 = \neg x_4 \vee x_5$

$c_9 = x_7 \vee x_5$ $\quad c_{10} = \neg x_7 \vee x_5$ $\quad c_{11} = \neg x_5 \vee x_3$ $\quad c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1, c_9\}$
- SAT($\mathcal{F} \setminus \{c_1, c_9\}$)? No
- Core of $\mathcal{F}$: $\{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_1, c_9\}$
- SAT($\mathcal{F} \setminus \{c_1, c_9\}$)? No
- Core of $\mathcal{F}$: $\{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}$. Update $\mathcal{K}$

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$:

MHS approach for MaxSAT

$$c_1 = x_6 \vee x_2 \qquad c_2 = \neg x_6 \vee x_2 \qquad c_3 = \neg x_2 \vee x_1 \qquad c_4 = \neg x_1$$

$$c_5 = \neg x_6 \vee x_8 \qquad c_6 = x_6 \vee \neg x_8 \qquad c_7 = x_2 \vee x_4 \qquad c_8 = \neg x_4 \vee x_5$$

$$c_9 = x_7 \vee x_5 \qquad c_{10} = \neg x_7 \vee x_5 \qquad c_{11} = \neg x_5 \vee x_3 \qquad c_{12} = \neg x_3$$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_4, c_9\}$

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$ $\quad c_2 = \neg x_6 \vee x_2$ $\quad c_3 = \neg x_2 \vee x_1$ $\quad c_4 = \neg x_1$

$c_5 = \neg x_6 \vee x_8$ $\quad c_6 = x_6 \vee \neg x_8$ $\quad c_7 = x_2 \vee x_4$ $\quad c_8 = \neg x_4 \vee x_5$

$c_9 = x_7 \vee x_5$ $\quad c_{10} = \neg x_7 \vee x_5$ $\quad c_{11} = \neg x_5 \vee x_3$ $\quad c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_4, c_9\}$
- SAT($\mathcal{F} \setminus \{c_4, c_9\}$)?

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_4, c_9\}$
- SAT($\mathcal{F} \setminus \{c_4, c_9\}$)? Yes

MHS approach for MaxSAT

$c_1 = x_6 \vee x_2$	$c_2 = \neg x_6 \vee x_2$	$c_3 = \neg x_2 \vee x_1$	$c_4 = \neg x_1$
$c_5 = \neg x_6 \vee x_8$	$c_6 = x_6 \vee \neg x_8$	$c_7 = x_2 \vee x_4$	$c_8 = \neg x_4 \vee x_5$
$c_9 = x_7 \vee x_5$	$c_{10} = \neg x_7 \vee x_5$	$c_{11} = \neg x_5 \vee x_3$	$c_{12} = \neg x_3$

$$\mathcal{K} = \{\{c_1, c_2, c_3, c_4\}, \{c_9, c_{10}, c_{11}, c_{12}\}, \{c_3, c_4, c_7, c_8, c_{11}, c_{12}\}\}$$

- Find MHS of $\mathcal{K}$: E.g. $\{c_4, c_9\}$
- SAT($\mathcal{F} \setminus \{c_4, c_9\}$)? Yes
- Terminate & return 2

MaxSAT solving with SAT oracles

- A sample of recent algorithms:

Algorithm	# Oracle Queries	Reference
Linear search SU	Exponential***	[e.g. LBP10]
Binary search	Linear*	[e.g. FM06]
FM/WMSU1/WPM1	Exponential**	[FM06,MSM08,MMSP09,ABL09a,ABGL12]
WPM2	Exponential**	[ABL10,ABGL13]
Bin-Core-Dis	Linear	[HMMS11,MHMS12]
Iterative MHS	Exponential	[DB11,DB13a,DB13b]

* $\mathcal{O}(\log m)$ queries with SAT oracle, for (partial) unweighted MaxSAT

** Weighted case; depends on computed cores

*** On # bits of problem instance (due to weights)

- But also additional recent work:
  - Progression
  - Soft cardinality constraints (OLL)
  - MaxSAT resolution
  - ...

Outline

Background

MaxSAT Solving

2QBF Solving

Abstraction refinement for QBF

- Many approaches proposed for solving QBF [...]
- Abstraction-refinement proposed for 2QBF in 2011 [JMS11]
- Extended to QBF in 2012 [JKMSC12]
- Significant impact in QBF competitions
- Influenced research in QBF solvers
  - E.g. see conference papers in 2015/2016

- **Ack:** Slides adapted from M. Janota SAT'11 talk

Problem definition

Given: $\exists X \forall Y . \phi$, where ϕ is a propositional formula

Question: Is there assignment ν to X variables such that $\forall Y . \phi[X/\nu]$?

Problem definition

Given: $\exists X \forall Y.\phi$, where ϕ is a propositional formula

Question: Is there assignment ν to X variables such that $\forall Y.\phi[X/\nu]$?

Example

$$\exists x_1, x_2\ \forall y_1, y_2.\ (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$$

solution: $x_1 = 0, x_2 = 0$

Problem definition

Given: $\exists X \forall Y . \phi$, where ϕ is a propositional formula

Question: Is there assignment ν to X variables such that $\forall Y . \phi[X/\nu]$?

Example

$$\exists x_1, x_2 \; \forall y_1, y_2 . \; (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$$

solution: $x_1 = 0, x_2 = 0$

A simple algorithm

- While **true**
  - Pick fresh assignment ν to X variables
  - Check with SAT solver whether $\forall Y . \phi[X/\nu]$ holds

Problem definition

Given: $\exists X \forall Y . \phi$, where ϕ is a propositional formula
Question: Is there assignment ν to X variables such that $\forall Y . \phi[X/\nu]$?

Example

$$\exists x_1, x_2 \; \forall y_1, y_2 . \; (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$$

solution: $x_1 = 0, x_2 = 0$

A simple algorithm

- While **true**
  - Pick fresh assignment ν to X variables
  - Check with SAT solver whether $\forall Y . \phi[X/\nu]$ holds
    - **How?**

Problem definition

Given: $\exists X \forall Y.\phi$, where ϕ is a propositional formula

Question: Is there assignment ν to X variables such that $\forall Y.\phi[X/\nu]$?

Example

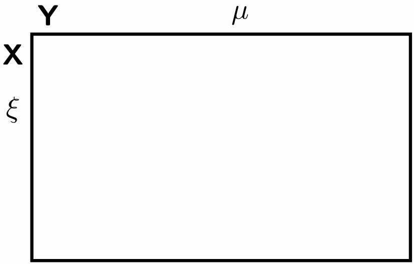
$$\exists x_1, x_2 \; \forall y_1, y_2. \; (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$$

solution: $x_1 = 0, x_2 = 0$

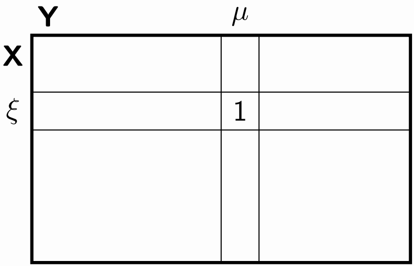
A simple algorithm

- While **true**
  - Pick fresh assignment ν to X variables
  - Check with SAT solver whether $\forall Y.\phi[X/\nu]$ holds
    - **How?** Check SAT$(\neg\phi[X/\nu])$ is unsat

Looking at assignments



Looking at assignments



Looking at assignments

	Y				μ		
X							
ξ	1	0	$\cdots$	0	1	$\cdots$	1
			$\cdots$			$\cdots$	

Looking at assignments

	Y				μ		
X							
ξ	1	0	$\cdots$	0	1	$\cdots$	1
			$\cdots$			$\cdots$	
ν	1	1	$\cdots$	1	1	$\cdots$	1

Looking at assignments

	Y				μ		
X							
ξ	1	0	$\cdots$	0	1	$\cdots$	1
			$\cdots$			$\cdots$	
ν	1	1	$\cdots$	1	1	$\cdots$	1
					$\phi[Y/\mu]$		

Expanding into SAT

$$\exists X \forall Y.\ \phi \implies \mathsf{SAT}\left(\bigwedge_{\mu \in \mathcal{B}^{|Y|}} \phi[Y/\mu]\right)$$

Expanding into SAT

$$\exists X \forall Y.\, \phi \implies \mathsf{SAT}\left(\bigwedge_{\mu \in \mathcal{B}^{|Y|}} \phi[Y/\mu] \right)$$

Example

$$\exists x_1, x_2\ \forall y_1, y_2.\ (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$$

$$\begin{array}{cc} & (x_1 \rightarrow 0) \wedge (x_2 \rightarrow 0) \\ \wedge & (x_1 \rightarrow 0) \wedge (x_2 \rightarrow 1) \\ \wedge & (x_1 \rightarrow 1) \wedge (x_2 \rightarrow 0) \\ \wedge & (x_1 \rightarrow 1) \wedge (x_2 \rightarrow 1) \end{array}$$

Abstraction

- Consider only some set of assignments $W \subseteq \mathcal{B}^{|Y|}$

$$\bigwedge_{\mu \in W} \phi[Y/\mu]$$

Abstraction

- Consider only some set of assignments $W \subseteq \mathcal{B}^{|Y|}$

$$\bigwedge_{\mu \in W} \phi[Y/\mu]$$

- A solution to the original problem is also a solution to the abstraction

$$\bigwedge_{\mu \in \mathcal{B}^{|Y|}} \phi[Y/\mu] \;\Rightarrow\; \bigwedge_{\mu \in W} \phi[Y/\mu]$$

Abstraction

- Consider only some set of assignments $W \subseteq \mathcal{B}^{|Y|}$

$$\bigwedge_{\mu \in W} \phi[Y/\mu]$$

- A solution to the original problem is also a solution to the abstraction

$$\bigwedge_{\mu \in \mathcal{B}^{|Y|}} \phi[Y/\mu] \;\Rightarrow\; \bigwedge_{\mu \in W} \phi[Y/\mu]$$

- But converse not true
  - A solution to an abstraction is not necessarily a solution to the original problem

CEGAR loop

input : $\exists X \forall Y . \phi$
output: (true, ν) if there exists ν s.t. $\forall Y \phi[X/\nu]$,
(false, –) otherwise

```
W ← ∅
while true do
    (outc₁, ν) ← SAT(⋀_{μ∈W} φ[Y/μ])          // find a candidate
    if outc₁ = false then
        return (false,–)                        // no candidate found
    end
    if ν is a solution                          // solution check
     then
        return (true, ν)
    else
        Grow W                                  // refinement
    end
end
```

CEGAR loop

input : $\exists X \forall Y . \phi$
output: (true, ν) if there exists ν s.t. $\forall Y \phi[X/\nu]$,
$(\text{false}, -)$ otherwise

```
W ← ∅
while true do
    (outc₁, ν) ← SAT(⋀_{μ∈W} φ[Y/μ])          // find a candidate
    if outc₁ = false then
        return (false,–)                       // no candidate found
    end
    if ν is a solution                         // solution check
     then
        return (true, ν)
    else
        Grow W                                 // refinement
    end
end
```

Checking for a solution

An assignment ν is a solution to $\exists X \forall Y.\phi$ iff

$$\forall Y.\phi[X/\nu] \quad \textit{iff} \quad \mathsf{UNSAT}(\neg\phi[X/\nu])$$

Checking for a solution

An assignment ν is a solution to $\exists X \forall Y.\phi$ iff

$$\forall Y.\phi[X/\nu] \quad \textit{iff} \quad \mathrm{UNSAT}(\neg\phi[X/\nu])$$

If $\mathrm{SAT}(\neg\phi[X/\nu])$ for some μ, then μ is a counterexample to ν

Checking for a solution

An assignment ν is a solution to $\exists X \forall Y . \phi$ iff

$$\forall Y . \phi[X/\nu] \;\; \textit{iff} \;\; \text{UNSAT}(\neg\phi[X/\nu])$$

If $\text{SAT}(\neg\phi[X/\nu])$ for some μ, then μ is a counterexample to ν

Example

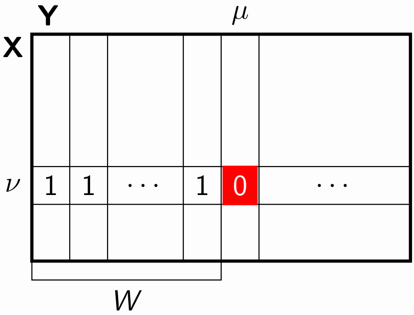
$\exists x_1, x_2 \; \forall y_1, y_2 . \; (x_1 \rightarrow y_1) \wedge (x_2 \rightarrow y_2)$

- candidate: $x_1 = 1, x_2 = 1$
- $\neg\phi[X/\nu] \triangleq \neg y_1 \vee \neg y_2$
- counterexamples: $\begin{array}{l} y_1 = 0, y_2 = 0 \\ y_1 = 0, y_2 = 1 \\ y_1 = 1, y_2 = 0 \end{array}$

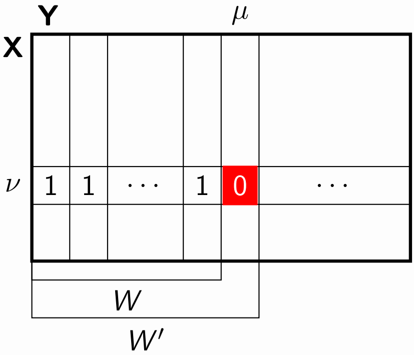
Refinement

	Y					
X						
ν	1	1	$\cdots$	1	1	$\cdots$

Refinement



Refinement

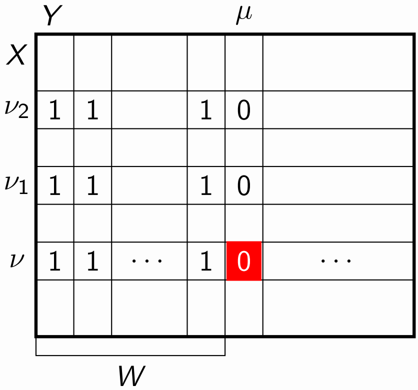


2QBF algorithm

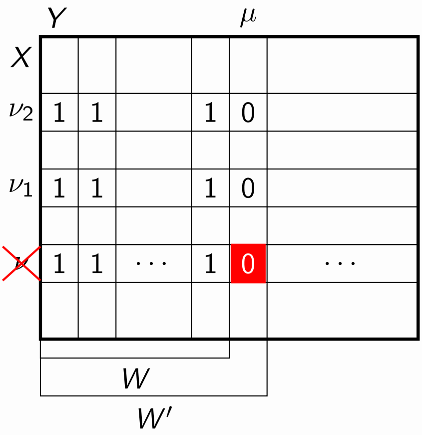
input : $\exists X \forall Y . \phi$
output: (true, ν) if there exists ν s.t. $\forall Y \phi[X/\nu]$,
$(\text{false}, -)$ otherwise

```
ω ← 1
while true do
    (outc₁, ν) ← SAT(ω)                    // find a candidate solution
    if outc₁ = false then
        return (false,–)                   // no candidate found
    end
    (outc₂, μ) ← SAT (¬φ[X/ν])             // find a counterexample
    if outc₂ = false then
        return (true, ν)                   // candidate is a solution
    end
    ω ← ω ∧ φ[Y/μ]                         // refine
end
```

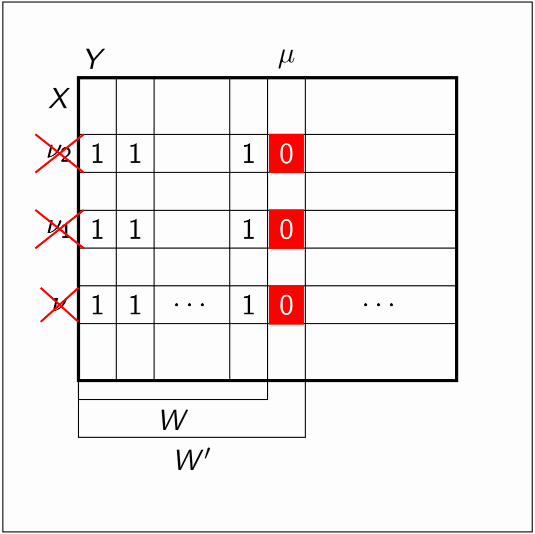
Properties of refinement



Properties of refinement



Properties of refinement



About refinement step

- No candidate for counterexample appears more than once
- Thus, upper bound on the number of iterations is:

$$\min\left\{2^{|X|}, 2^{|Y|}\right\}$$

About refinement step

- No candidate for counterexample appears more than once
- Thus, upper bound on the number of iterations is:

$$\min\left\{2^{|X|}, 2^{|Y|}\right\}$$

- Heuristic: look for such counterexamples that are also counterexamples to many other candidates, look for μ s.t.

$$\neg\phi[X/\nu] \wedge \max\left(|\{\nu' \mid \neg\phi[X/\nu', Y/\mu]\}|\right)$$

Part II

Back Again (to NP)

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)

- What we have been looking at?
  - Reduce problems to one concrete problem, i.e. Horn MaxSAT
  - Develop fast algorithms for Horn MaxSAT
    - Use IHSes, MHSes, MUSes, etc.

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)

- What we have been looking at?
  - Reduce problems to one concrete problem, i.e. Horn MaxSAT
  - Develop fast algorithms for Horn MaxSAT
    - Use IHSes, MHSes, MUSes, etc.

- What we found out?
  - Reductions are remarkably effective for PHP in practice

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)
- What we have been looking at?
  - Reduce problems to one concrete problem, i.e. Horn MaxSAT
  - Develop fast algorithms for Horn MaxSAT
    - Use IHSes, MHSes, MUSes, etc.
- What we found out?
  - Reductions are remarkably effective for PHP in practice
  - There exist **polynomial time proofs** that PHP is unsatisfiable **!**

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)

- What we have been looking at?
  - Reduce problems to one concrete problem, i.e. Horn MaxSAT
  - Develop fast algorithms for Horn MaxSAT
    - Use IHSes, MHSes, MUSes, etc.

- What we found out?
  - Reductions are remarkably effective for PHP in practice
  - There exist **polynomial time proofs** that PHP is unsatisfiable **!**
    - Using **core-guided algorithms**; and
    - Using **MaxSAT resolution**

Why back again to NP?

- **Fact:** There are many hard examples for resolution and CDCL
  - One example are pigeonhole formulas (PHP) (more later)

- What we have been looking at?
  - Reduce problems to one concrete problem, i.e. Horn MaxSAT
  - Develop fast algorithms for Horn MaxSAT
    - Use IHSes, MHSes, MUSes, etc.

- What we found out?
  - Reductions are remarkably effective for PHP in practice
  - There exist **polynomial time proofs** that PHP is unsatisfiable **!**
    - Using **core-guided algorithms**; and
    - Using **MaxSAT resolution**
  - **But**, **core-guided algorithms also use CDCL !**
  - **Also**, **MHS MaxSAT algorithms are effective on hard problems**

Plan for part B

1. Recap PHP
2. Reduce SAT to Horn MaxSAT
   - Also, what happens to PHP?
3. Develop polynomial time proofs of the unsatisfiability of PHP
   - Using an MSU3-like MaxSAT algorithm
   - Using MaxSAT resolution
4. Experimental results
   - PHP, Urquhart, and combinations thereof
5. Detailed description available from:

   https://arxiv.org/abs/1705.01477

Outline

Pigeonhole Formulas

Reduction: SAT to Horn MaxSAT

Polynomial Time Proofs

Experimental Results

Pigeonhole formulas I

- Pigeonhole principle:
  - Typical: if $m+1$ pigeons are distributed by m holes, then at least one hole contains more than one pigeon
  - Alternative: there exists no injective function mapping from $\{1, 2, ..., m+1\}$ to $\{1, 2, ..., m\}$, for $m \geq 1$

Pigeonhole formulas I

- Pigeonhole principle:
  - Typical: if $m+1$ pigeons are distributed by m holes, then at least one hole contains more than one pigeon
  - Alternative: there exists no injective function mapping from $\{1, 2, ..., m+1\}$ to $\{1, 2, ..., m\}$, for $m \geq 1$

- Propositional formulation:

  Does there exist assignment such that the $m+1$ pigeons can be placed into m holes?

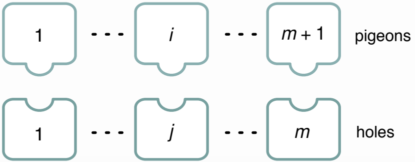
Pigeonhole formulas I

- Pigeonhole principle:
  - Typical: if $m+1$ pigeons are distributed by m holes, then at least one hole contains more than one pigeon
  - Alternative: there exists no injective function mapping from $\{1, 2, ..., m+1\}$ to $\{1, 2, ..., m\}$, for $m \geq 1$

- Propositional formulation:

  Does there exist assignment such that the $m+1$ pigeons can be placed into m holes?

- Encoding: x_{ij} variables



Pigeonhole formulas II – propositional encoding PHP_m^{m+1}

- Variables:
  - $x_{ij} = 1$ iff the i^{th} pigeon is placed in the j^{th} hole, $1 \leq i \leq m+1$, $1 \leq j \leq m$

Pigeonhole formulas II – propositional encoding PHP_m^{m+1}

- Variables:
  - $x_{ij} = 1$ iff the i^{th} pigeon is placed in the j^{th} hole, $1 \leq i \leq m+1$, $1 \leq j \leq m$
- Constraints:
  - Each pigeon must be placed in at least one hole, and each hole must not have more than one pigeon

$$\bigwedge_{i=1}^{m+1} \mathrm{AtLeast1}(x_{i1}, \ldots, x_{im}) \wedge \bigwedge_{j=1}^{m} \mathrm{AtMost1}(x_{1j}, \ldots, x_{m+1\,j})$$

Pigeonhole formulas II – propositional encoding PHP_m^{m+1}

- Variables:
  - $x_{ij} = 1$ iff the i^{th} pigeon is placed in the j^{th} hole, $1 \leq i \leq m+1$, $1 \leq j \leq m$
- Constraints:
  - Each pigeon must be placed in at least one hole, and each hole must not have more than one pigeon

$$\bigwedge_{i=1}^{m+1} \text{AtLeast1}(x_{i1}, \ldots, x_{im}) \wedge \bigwedge_{j=1}^{m} \text{AtMost1}(x_{1j}, \ldots, x_{m+1\,j})$$

- Example encoding, with pairwise encoding for AtMost1 constraint:

Constraint	Clause(s)
$\wedge_{i=1}^{m+1}\text{AtLeast1}(x_{i1}, \ldots, x_{im})$	$(x_{i1} \vee \ldots \vee x_{im})$
$\wedge_{j=1}^{m}\text{AtMost1}(x_{1j}, \ldots, x_{m+1\,j})$	$\wedge_{r=2}^{m+1} \wedge_{s=1}^{r-1} (\neg x_{rj} \vee \neg x_{sj})$

Outline

Pigeonhole Formulas

Reduction: SAT to Horn MaxSAT

Polynomial Time Proofs

Experimental Results

Reducing SAT to Horn MaxSAT

- Formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_t\}$

Reducing SAT to Horn MaxSAT

- Formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_t\}$
- Replace each original variable $x_i \in X$ by n_i and p_i, s.t.
  - $n_i = 1$ iff $x_i = 0$
  - $p_i = 1$ iff $x_i = 1$
  - Add (hard Horn) constraint $(\neg n_i \vee \neg p_i)$ $\Leftarrow$ set of clauses $\mathcal{P}$

Reducing SAT to Horn MaxSAT

- Formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_t\}$
- Replace each original variable $x_i \in X$ by n_i and p_i, s.t.
  - $n_i = 1$ iff $x_i = 0$
  - $p_i = 1$ iff $x_i = 1$
  - Add (hard Horn) constraint $(\neg n_i \vee \neg p_i)$ $\Leftarrow$ set of clauses $\mathcal{P}$
- Translate each clause $c_r \in \mathcal{F}$ into (hard Horn) clause $c'_r \in \mathcal{F}_H$:
  - Literal x_i converted to $\neg n_i$
  - Literal $\neg x_i$ converted to $\neg p_i$
  - Resulting clause is goal clause $\Leftarrow$ (can do better)

Reducing SAT to Horn MaxSAT

- Formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_t\}$
- Replace each original variable $x_i \in X$ by n_i and p_i, s.t.
  - $n_i = 1$ iff $x_i = 0$
  - $p_i = 1$ iff $x_i = 1$
  - Add (hard Horn) constraint $(\neg n_i \vee \neg p_i)$ $\Leftarrow$ set of clauses $\mathcal{P}$
- Translate each clause $c_r \in \mathcal{F}$ into (hard Horn) clause $c'_r \in \mathcal{F}_H$:
  - Literal x_i converted to $\neg n_i$
  - Literal $\neg x_i$ converted to $\neg p_i$
  - Resulting clause is goal clause $\Leftarrow$ (can do better)
- Soft clauses: $\mathcal{S} = \{(n_1), \ldots, (n_t), (p_1), \ldots, (p_t)\}$
- Horn MaxSAT formula: $\langle \mathcal{F}_H \cup \mathcal{P}, \mathcal{S} \rangle$

Reducing SAT to Horn MaxSAT

- Formula $\mathcal{F}$ with variables $X = \{x_1, \ldots, x_t\}$
- Replace each original variable $x_i \in X$ by n_i and p_i, s.t.
  - $n_i = 1$ iff $x_i = 0$
  - $p_i = 1$ iff $x_i = 1$
  - Add (hard Horn) constraint $(\neg n_i \vee \neg p_i)$ $\Leftarrow$ set of clauses $\mathcal{P}$
- Translate each clause $c_r \in \mathcal{F}$ into (hard Horn) clause $c'_r \in \mathcal{F}_H$:
  - Literal x_i converted to $\neg n_i$
  - Literal $\neg x_i$ converted to $\neg p_i$
  - Resulting clause is goal clause $\Leftarrow$ (can do better)
- Soft clauses: $\mathcal{S} = \{(n_1), \ldots, (n_t), (p_1), \ldots, (p_t)\}$
- Horn MaxSAT formula: $\langle \mathcal{F}_H \cup \mathcal{P}, \mathcal{S} \rangle$
- **Claim**:

  $\mathcal{F}$ is SAT iff Horn MaxSAT formula has solution with cost $\leq t$
  - There exists assignment that satisfies hard clauses $\mathcal{F}_H$ and at least t soft clauses from $\mathcal{S}$, i.e. cost $\leq t$
  - Due to $\mathcal{P}$ clauses, cost $\geq t$; thus $\mathcal{F}$ is SAT iff cost $= t$

An example

- CNF formula:

$$\mathcal{F} = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

An example

- CNF formula:

$$\mathcal{F} = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

- New variables: $\{n_1, p_1, n_2, p_2, n_3, p_3\}$
- Soft clauses: $\mathcal{S} = \{(n_1), (p_1), (n_2), (p_2), (n_3), (p_3)\}$

An example

- CNF formula:

$$\mathcal{F} = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

- New variables: $\{n_1, p_1, n_2, p_2, n_3, p_3\}$
- Soft clauses: $\mathcal{S} = \{(n_1), (p_1), (n_2), (p_2), (n_3), (p_3)\}$
- Clauses in $\mathcal{P}$:

$$\mathcal{P} \triangleq (\neg n_1 \vee \neg p_1) \wedge (\neg n_2 \vee \neg p_2) \wedge (\neg n_3 \vee \neg p_3)$$

An example

- CNF formula:
$$\mathcal{F} = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$
- New variables: $\{n_1, p_1, n_2, p_2, n_3, p_3\}$
- Soft clauses: $\mathcal{S} = \{(n_1), (p_1), (n_2), (p_2), (n_3), (p_3)\}$
- Clauses in $\mathcal{P}$:
$$\mathcal{P} \triangleq (\neg n_1 \vee \neg p_1) \wedge (\neg n_2 \vee \neg p_2) \wedge (\neg n_3 \vee \neg p_3)$$
- Original clauses converted to:
$$\mathcal{F}_H \triangleq (\neg n_1 \vee \neg p_2 \vee \neg n_3) \wedge (\neg n_2 \vee \neg n_3) \wedge (\neg p_1 \vee \neg p_3)$$

An example

- CNF formula:

$$\mathcal{F} = (x_1 \vee \neg x_2 \vee x_3) \wedge (x_2 \vee x_3) \wedge (\neg x_1 \vee \neg x_3)$$

- New variables: $\{n_1, p_1, n_2, p_2, n_3, p_3\}$
- Soft clauses: $\mathcal{S} = \{(n_1), (p_1), (n_2), (p_2), (n_3), (p_3)\}$
- Clauses in $\mathcal{P}$:

$$\mathcal{P} \triangleq (\neg n_1 \vee \neg p_1) \wedge (\neg n_2 \vee \neg p_2) \wedge (\neg n_3 \vee \neg p_3)$$

- Original clauses converted to:

$$\mathcal{F}_H \triangleq (\neg n_1 \vee \neg p_2 \vee \neg n_3) \wedge (\neg n_2 \vee \neg n_3) \wedge (\neg p_1 \vee \neg p_3)$$

- Resulting formula: $\langle \mathcal{F}_H \cup \mathcal{P}, \mathcal{S} \rangle$
- $\mathcal{F}$ is satisfiable iff Horn MaxSAT formula has a solution with cost 3

PHP as Horn MaxSAT

- New variables n_{ij} and p_{ij}, for each x_{ij}, $1 \leq i \leq m+1, 1 \leq j \leq m$
- The soft clauses $\mathcal{S}$, with $|\mathcal{S}| = 2m(m+1)$, are given by

$$\begin{array}{ll} \{ & (n_{11}), \ldots, (n_{1m}), \ldots, (n_{m+1\,1}), \ldots, (n_{m+1\,m}), \\ & (p_{11}), \ldots, (p_{1m}), \ldots, (p_{m+1\,1}), \ldots, (p_{m+1\,m}) \quad \} \end{array}$$

PHP as Horn MaxSAT

- New variables n_{ij} and p_{ij}, for each x_{ij}, $1 \leq i \leq m+1, 1 \leq j \leq m$
- The soft clauses $\mathcal{S}$, with $|\mathcal{S}| = 2m(m+1)$, are given by

$$\begin{array}{ll} \{ & (n_{11}), \ldots, (n_{1m}), \ldots, (n_{m+1\,1}), \ldots, (n_{m+1\,m}), \\ & (p_{11}), \ldots, (p_{1m}), \ldots, (p_{m+1\,1}), \ldots, (p_{m+1\,m}) \quad \} \end{array}$$

- Clauses in $\mathcal{P}$: $\mathcal{P} = \{(\neg n_{ij} \vee \neg p_{ij}) \,|\, 1 \leq i \leq m+1, 1 \leq j \leq m\}$

PHP as Horn MaxSAT

- New variables n_{ij} and p_{ij}, for each x_{ij}, $1 \leq i \leq m+1, 1 \leq j \leq m$
- The soft clauses $\mathcal{S}$, with $|\mathcal{S}| = 2m(m+1)$, are given by

$$\begin{array}{ll} \{ & (n_{11}), \ldots, (n_{1m}), \ldots, (n_{m+1\,1}), \ldots, (n_{m+1\,m}), \\ & (p_{11}), \ldots, (p_{1m}), \ldots, (p_{m+1\,1}), \ldots, (p_{m+1\,m}) \quad \} \end{array}$$

- Clauses in $\mathcal{P}$: $\mathcal{P} = \{(\neg n_{ij} \vee \neg p_{ij}) \,|\, 1 \leq i \leq m+1, 1 \leq j \leq m\}$
- AtLeast1 constraints encoded as $\mathcal{L}_i, 1 \leq i \leq m+1$
- AtMost1 constraints encoded as $\mathcal{M}_j, 1 \leq j \leq m$

PHP as Horn MaxSAT

- New variables n_{ij} and p_{ij}, for each x_{ij}, $1 \leq i \leq m+1, 1 \leq j \leq m$
- The soft clauses $\mathcal{S}$, with $|\mathcal{S}| = 2m(m+1)$, are given by

$$\begin{array}{ll}\{ & (n_{11}), \ldots, (n_{1m}), \ldots, (n_{m+1\,1}), \ldots, (n_{m+1\,m}), \\ & (p_{11}), \ldots, (p_{1m}), \ldots, (p_{m+1\,1}), \ldots, (p_{m+1\,m}) \quad \}\end{array}$$

- Clauses in $\mathcal{P}$: $\mathcal{P} = \{(\neg n_{ij} \vee \neg p_{ij}) \,|\, 1 \leq i \leq m+1, 1 \leq j \leq m\}$
- AtLeast1 constraints encoded as $\mathcal{L}_i, 1 \leq i \leq m+1$
- AtMost1 constraints encoded as $\mathcal{M}_j, 1 \leq j \leq m$
- Full reduction of PHP to Horn MaxSAT

$$\langle \mathcal{H}, \mathcal{S} \rangle = \left\langle \wedge_{i=1}^{m+1} \mathcal{L}_i \wedge \wedge_{j=1}^{m} \mathcal{M}_j \wedge \mathcal{P}, \mathcal{S} \right\rangle$$

PHP as Horn MaxSAT

- New variables n_{ij} and p_{ij}, for each x_{ij}, $1 \leq i \leq m+1, 1 \leq j \leq m$
- The soft clauses $\mathcal{S}$, with $|\mathcal{S}| = 2m(m+1)$, are given by

$$\begin{array}{ll} \{ & (n_{11}), \ldots, (n_{1m}), \ldots, (n_{m+1\,1}), \ldots, (n_{m+1\,m}), \\ & (p_{11}), \ldots, (p_{1m}), \ldots, (p_{m+1\,1}), \ldots, (p_{m+1\,m}) \quad \} \end{array}$$

- Clauses in $\mathcal{P}$: $\mathcal{P} = \{(\neg n_{ij} \vee \neg p_{ij}) \,|\, 1 \leq i \leq m+1, 1 \leq j \leq m\}$
- AtLeast1 constraints encoded as $\mathcal{L}_i, 1 \leq i \leq m+1$
- AtMost1 constraints encoded as $\mathcal{M}_j, 1 \leq j \leq m$
- Full reduction of PHP to Horn MaxSAT

$$\langle \mathcal{H}, \mathcal{S} \rangle = \left\langle \wedge_{i=1}^{m+1} \mathcal{L}_i \wedge \wedge_{j=1}^{m} \mathcal{M}_j \wedge \mathcal{P}, \mathcal{S} \right\rangle$$

- **No** more than $m(m+1)$ clauses can be satisfied, due to $\mathcal{P}$
- PHP_m^{m+1} is satisfiable iff there exists an assignment that satisfies the hard clauses $\mathcal{H}$ and $m(m+1)$ soft clauses from $\mathcal{S}$

PHP as Horn MaxSAT II

- Clauses in each $\mathcal{L}_i$ and in each $\mathcal{M}_j$, with pairwise encoding

Original Constraint	Encoded To	Clauses
$\wedge_{i=1}^{m+1}\text{AtLeast1}(x_{i1},\ldots,x_{im})$	$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$
$\wedge_{j=1}^{m}\text{AtMost1}(x_{1j},\ldots,x_{m+1,j})$	$\mathcal{M}_j$	$\wedge_{r=2}^{m+1} \wedge_{s=1}^{r-1} (\neg p_{rj} \vee \neg p_{sj})$

PHP as Horn MaxSAT II

- Clauses in each $\mathcal{L}_i$ and in each $\mathcal{M}_j$, with pairwise encoding

Original Constraint	Encoded To	Clauses
$\wedge_{i=1}^{m+1}\text{AtLeast1}(x_{i1},\ldots,x_{im})$	$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$
$\wedge_{j=1}^{m}\text{AtMost1}(x_{1j},\ldots,x_{m+1,j})$	$\mathcal{M}_j$	$\wedge_{r=2}^{m+1}\wedge_{s=1}^{r-1}(\neg p_{rj} \vee \neg p_{sj})$

- **Note:** constraints with key structural properties:

Constraint	Variables
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$
$\mathcal{L}_k$	$(\neg n_{k1} \vee \ldots \vee \neg n_{km})$
$\mathcal{M}_j$	$\wedge_{r=2}^{m+1}\wedge_{s=1}^{r-1}(\neg p_{rj} \vee \neg p_{sj})$
$\mathcal{M}_l$	$\wedge_{r=2}^{m+1}\wedge_{s=1}^{r-1}(\neg p_{rl} \vee \neg p_{sl})$

  - Variables in each $\mathcal{L}_i$ disjoint from **any** other $\mathcal{L}_k$ and $\mathcal{M}_j$, $k \neq i$
  - Variables in each $\mathcal{M}_j$ disjoint from **any** other $\mathcal{M}_l$, $l \neq j$

Outline

Pigeonhole Formulas

Reduction: SAT to Horn MaxSAT

Polynomial Time Proofs

Experimental Results

Main claims

Claim 1

Core-guided MaxSAT produces a lower bound on the number of falsified clauses of $\geq m(m+1)+1$ in polynomial time

Main claims

Claim 1

Core-guided MaxSAT produces a lower bound on the number of falsified clauses of $\geq m(m+1)+1$ in polynomial time

Claim 2

MaxSAT resolution produces a lower bound on the number of falsified clauses of $\geq m(m+1)+1$ in polynomial time

Main claims

Claim 1

Core-guided MaxSAT produces a lower bound on the number of falsified clauses of $\geq m(m+1)+1$ in polynomial time

Claim 2

MaxSAT resolution produces a lower bound on the number of falsified clauses of $\geq m(m+1)+1$ in polynomial time

Corollary

Horn MaxSAT encoding enables polynomial time proofs of the unsatisfiability of PHP instances, using CDCL SAT solvers

Proof of claim 1 – outline

1. Assume MSU3 MaxSAT algorithm
   - **Note:** Suffices to analyze disjoint sets separately

Proof of claim 1 – outline

1. Assume MSU3 MaxSAT algorithm
   – **Note:** Suffices to analyze disjoint sets separately

2. Relate soft clauses with each $\mathcal{L}_i$ and each $\mathcal{M}_j$
   – Each constraint disjoint from the others (but not from $\mathcal{P}$)

Proof of claim 1 – outline

1. Assume MSU3 MaxSAT algorithm
   - **Note:** Suffices to analyze disjoint sets separately
2. Relate soft clauses with each $\mathcal{L}_i$ and each $\mathcal{M}_j$
   - Each constraint disjoint from the others (but not from $\mathcal{P}$)
3. Derive large enough lower bound on # of falsified clauses:

Constr. type	# falsified cls	# constr	In total
$\mathcal{L}_i$	1	$i = 1, \ldots, m+1$	$m+1$
$\mathcal{M}_j$	m	$j = 1, \ldots, m$	$m \cdot m$
			$m(m+1)+1$

Proof of claim 1 – outline

1. Assume MSU3 MaxSAT algorithm
   - **Note:** Suffices to analyze disjoint sets separately
2. Relate soft clauses with each $\mathcal{L}_i$ and each $\mathcal{M}_j$
   - Each constraint disjoint from the others (but not from $\mathcal{P}$)
3. Derive large enough lower bound on # of falsified clauses:

Constr. type	# falsified cls	# constr	In total
$\mathcal{L}_i$	1	$i = 1, \ldots, m+1$	$m+1$
$\mathcal{M}_j$	m	$j = 1, \ldots, m$	$m \cdot m$
			$m(m+1)+1$

4. Each increase in the value of the lower bound obtained by unit propagation (UP)
   - In total: polynomial number of (linear time) UP runs

Proof of claim 1 – unit propagation steps I

Constr	Hard cls	Soft cls	Relaxed clauses	Updated AtMost$k$ constr	LB incr
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(s_{il} \vee n_{i1})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} s_{il} \leq 1$	1
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{2j})$	$(p_{1j}), (p_{2j})$	$(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$	$\sum_{l=1}^{2} r_{lj} \leq 1$	1
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{3j})$, $(\neg p_{2j} \vee \neg p_{3j})$, $(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$, $\sum_{l=1}^{2} r_{lj} \leq 1$	(p_{3j})	$(r_{3j} \vee p_{3j})$	$\sum_{l=1}^{3} r_{lj} \leq 2$	1
		$\ldots$			
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{m+1j}), \ldots$, $(\neg p_{mj} \vee \neg p_{m+1j})$, $(r_{1j} \vee p_{1j}), \ldots$, $(r_{mj} \vee p_{mj})$, $\sum_{l=1}^{m} r_{lj} \leq m-1$	(p_{m+1j})	$(r_{m+1j} \vee p_{m+1j})$	$\sum_{l=1}^{m+1} r_{lj} \leq m$	1

Proof of claim 1 – unit propagation steps II

Clauses	Unit Propagation
$(p_{k+1\,j})$	$p_{k+1\,j} = 1$
$(\neg p_{1j} \vee \neg p_{k+1\,j}), \ldots, (\neg p_{kj} \vee \neg p_{k+1\,j})$	$p_{1j} = \ldots = p_{kj} = 0$
$(r_{1j} \vee p_{1j}), \ldots, (r_{kj} \vee p_{kj})$	$r_{1j} = \ldots = r_{kj} = 1$
$\sum_{l=1}^{k} r_{lj} \leq k - 1$	$\left(\sum_{l=1}^{k} r_{lj} \leq k - 1\right) \vdash_1 \bot$

- Key points:
  - For each $\mathcal{L}_i$, UP raises LB by 1
  - For each $\mathcal{M}_j$, UP raises LB by m
  - In total, UP raises LB by $m(m+1)+1$
  - PHP_m^{m+1} is unsatisfiable

Proof of claim 2 – recap MaxSAT resolution

- Clauses: $(x \vee A, u)$ and $(\neg x \vee B, w)$
- $m \triangleq \min(u, w)$
- $u \ominus w \triangleq (u == \top) \,?\, \top : u - w$, with $u \geq w$
- Example MaxSAT resolution steps:

Clause 1	Clause 2	Derived Clauses
$(x \vee A, u)$	$(\neg x \vee B, w)$	$(A \vee B, m)$, $(x \vee A, u \ominus m)$, $(\neg x \vee B, w \ominus m)$, $(x \vee A \vee \neg B, m)$, $(\neg x \vee \neg A \vee B, m)$
$(x \vee A, 1)$	$(\neg x, \top)$	$(A, 1)$, $(\neg x, \top)$, $(\neg x \vee \neg A, 1)$

Proof of claim 2 – outline

- Follow ideas used in previous proof
- Mimic unit propagation steps as MaxSAT resolution steps
- Each increase in LB corresponds to deriving one empty clause
- In total: polynomial number of steps, each running in polynomial time

Proof of claim 2 – key steps I

Constraint	Clauses	Resulting clause(s)
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im}, \top)$, $(n_{i1}, 1)$	$\boxed{(\neg n_{i2} \vee \ldots \vee \neg n_{im}, 1)}, \ldots$
$\mathcal{L}_i$	$(\neg n_{i2} \vee \ldots \vee \neg n_{im}, 1)$, $(n_{i2}, 1)$	$\boxed{(\neg n_{i3} \vee \ldots \vee \neg n_{im}, 1)}, \ldots$
	$\ldots$	
$\mathcal{L}_i$	$(\neg n_{im}, 1)$, $(n_{im}, 1)$	$\boxed{(\bot, 1)}, \ldots$
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{2j}, \top)$, $(p_{1j}, 1)$	$(\neg p_{2j}, 1)$, $(\neg p_{1j} \vee \neg p_{2j}, \top)$, $\boxed{(p_{1j} \vee p_{2j}, 1)}$
$\mathcal{M}_j$	$(\neg p_{2j}, 1)$, $(p_{2j}, 1)$	$\boxed{(\bot, 1)}$
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{3j}, \top)$, $(p_{1j} \vee p_{2j}, 1)$	$\boxed{(p_{2j} \vee \neg p_{3j}, 1)}$, $(\neg p_{1j} \vee \neg p_{3j}, \top)$, $(\neg p_{1j} \vee \neg p_{3j} \vee \neg p_{2j}, 1)$, $\boxed{(p_{1j} \vee p_{2j} \vee p_{3j}, 1)}$
$\mathcal{M}_j$	$(\neg p_{2j} \vee \neg p_{3j}, \top)$, $(p_{2j} \vee \neg p_{3j}, 1)$	$\boxed{(\neg p_{3j}, 1)}$, $(\neg p_{2j} \vee \neg p_{3j}, \top)$
$\mathcal{M}_j$	$(\neg p_{3j}, 1)$, $(p_{3j}, 1)$	$\boxed{(\bot, 1)}$

Proof of claim 2 – key steps II

Constraint	Clauses	Resulting clause(s)
	$\dots$	
$\mathcal{M}_j$	$(\neg p_{1j} \vee \neg p_{m+1j}, \top)$, $(p_{1j} \vee \ldots \vee p_{mj}, 1)$	$(p_{2j} \ldots p_{mj} \vee \neg p_{m+1j}, 1)$, $\ldots$
$\mathcal{M}_j$	$(\neg p_{2j} \vee \neg p_{m+1j}, \top)$, $(p_{2j} \vee \ldots \vee p_{mj} \vee \neg p_{m+1j}, 1)$	$(p_{3j} \ldots p_{mj} \vee \neg p_{m+1j}, 1)$, $\ldots$
	$\dots$	
$\mathcal{M}_j$	$(\neg p_{mj} \vee \neg p_{m+1j}, \top)$, $(p_{mj} \vee \neg p_{m+1j}, 1)$	$\neg p_{m+1j}, 1)$, $\ldots$
$\mathcal{M}_j$	$(p_{m+1j}, 1)$, $(\neg p_{m+1j}, 1)$	$(\perp, 1)$

- Key points:
  - For each $\mathcal{L}_i$, derive 1 empty clause
  - For each $\mathcal{M}_j$, derive m empty clauses
  - In total, derive $m(m+1)+1$ empty clauses
  - PHP_m^{m+1} is unsatisfiable

Outline

Pigeonhole Formulas

Reduction: SAT to Horn MaxSAT

Polynomial Time Proofs

Experimental Results

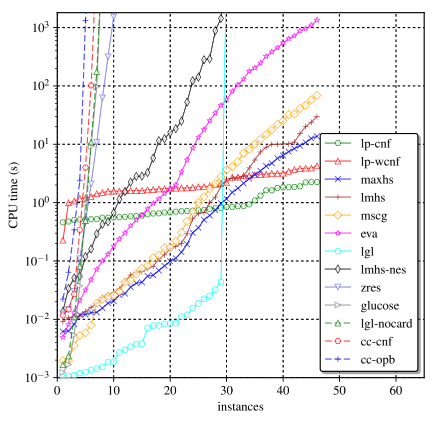
Experimental setup

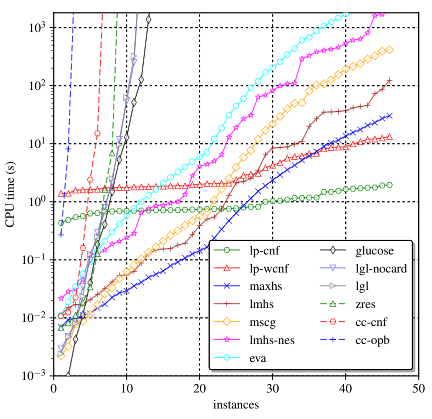
- Instances:
  - PHP-pw (46), PHP-sc (46), Urquhart (84), Comb (96)

- Solvers:

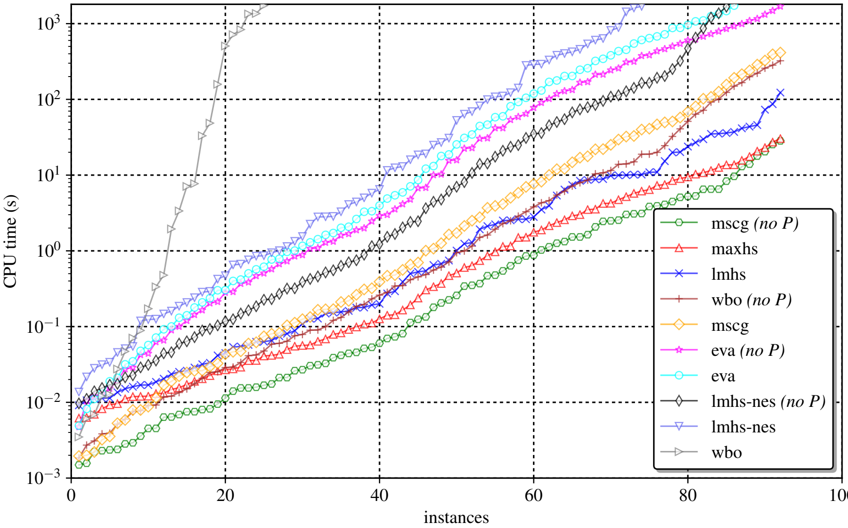
SAT		SAT+		IHS MaxSAT		CG MaxSAT			MRes	MIP	OPB		BDD
minisat	*glucose*	*lgl*	crypto	*maxhs*	*lmhs*	*mscg*	*wbo*	wpm3	*eva*	*lp*	*cc*	sat4j*	*zres*

Results on PHP instances: pw vs. sc

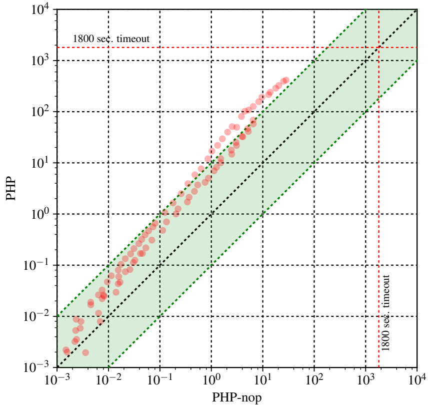


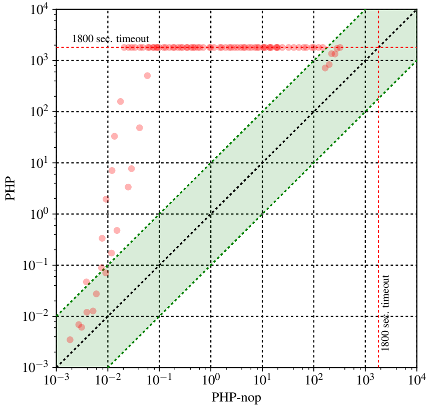


Effect of $\mathcal{P}$ clauses

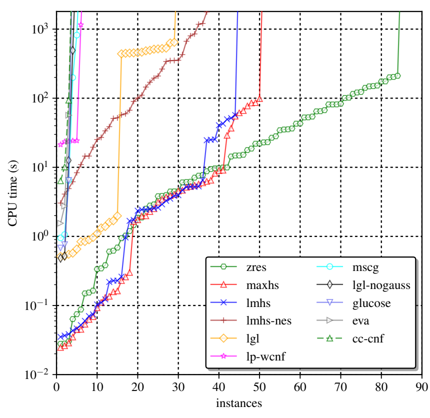


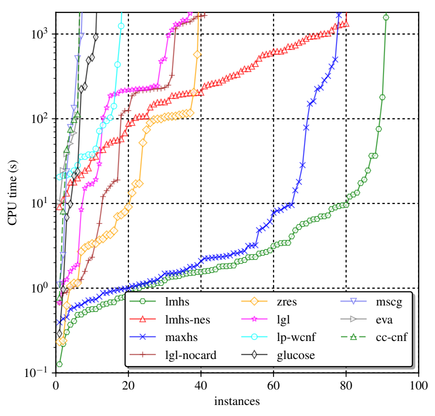
Effect of $\mathcal{P}$ clauses on mscg and wbo





Results on Urquhart & combined instances





More detail in arXiv report

"On Tackling the Limits of Resolution in SAT Solving"

A. Ignatiev, A. Morgado, and J. Marques-Silva

https://arxiv.org/abs/1705.01477

Part III

Wrap Up

Conclusions

- Covered some examples of problem solving using SAT oracles
  - MaxSAT solving
  - 2QBF solving
- But, many more examples:
  - MUS & MCS extraction
  - MUS & MCS enumeration
  - Prime compilation
  - Implicit hitting sets
  - Quantification: decision, QMaxSAT, abduction, ...
  - Smallest MUSes
  - Approximate model counting
  - Also: backbones; autarkies/lean kernels, ...
  - Also: (many) practical applications

Conclusions

- Covered some examples of problem solving using SAT oracles
  - MaxSAT solving
  - 2QBF solving
- But, many more examples:
  - MUS & MCS extraction
  - MUS & MCS enumeration
  - Prime compilation
  - Implicit hitting sets
  - Quantification: decision, QMaxSAT, abduction, ...
  - Smallest MUSes
  - Approximate model counting
  - Also: backbones; autarkies/lean kernels, ...
  - Also: (many) practical applications
- (Horn) MaxSAT solvers can solve (in polynomial time) hard instances for resolution
  - If equipped with the right reduction

Some research topics

- Beyond NP:
  - Query complexity
  - Enumeration
  - Quantification
  - Implicit hitting sets & duality
  - ...

Some research topics

- Beyond NP:
  - Query complexity
  - Enumeration
  - Quantification
  - Implicit hitting sets & duality
  - ...
- Applications:
  - Diagnosis
  - Axiom pinpointing
  - Planning
  - Reachability
  - Synthesis
  - Networking
  - Configuration
  - Argumentation
  - ...

Some research topics

- Beyond NP:
  - Query complexity
  - Enumeration
  - Quantification
  - Implicit hitting sets & duality
  - ...
- Applications:
  - Diagnosis
  - Axiom pinpointing
  - Planning
  - Reachability
  - Synthesis
  - Networking
  - Configuration
  - Argumentation
  - ...
- Also, where to go with Horn MaxSAT?

Thank You