On Incremental Core-Guided MaxSAT Solving

X. Si[1], X. Zhang[1], V. Manquinho[2], M. Janota[3], **A. Ignatiev**[4,5], and M. Naik[1]

September 6, 2016

[1] Georgia Institute of Technology, USA
[2] INESC-ID, IST, University of Lisbon, Portugal
[3] Microsoft Research, Cambridge, UK
[4] LaSIGE, FC, University of Lisbon, Portugal
[5] ISDCT SB RAS, Irkutsk, Russia

Maximum satisfiability

Maximum satisfiability

$$\text{given } F = F_{hard} \wedge F_{soft} \models \bot,$$

Maximum satisfiability

given $F = F_{hard} \wedge F_{soft} \models \bot$,

satisfy F_{hard} and maximize $\sum_{c \in F_{soft}} weight(c)$

Maximum satisfiability

$$\text{given } F = F_{hard} \wedge F_{soft} \models \bot,$$

$$\text{satisfy } F_{hard} \text{ and maximize } \sum_{c \in F_{soft}} weight(c)$$

$$\begin{array}{rcccc} F_{hard} & = & (\neg x \vee \neg y) & (\neg x \vee \neg z) & (\neg y \vee \neg z) \\ F_{soft} & = & (10, x) & (20, y) & (40, z) \end{array}$$

Maximum satisfiability

given $F = F_{hard} \wedge F_{soft} \models \bot$,

satisfy F_{hard} and maximize $\sum_{c \in F_{soft}} weight(c)$

$$
\begin{array}{lcccc}
F_{hard} & = & (\neg x \vee \neg y) & (\neg x \vee \neg z) & (\neg y \vee \neg z) \\
F_{soft} & = & (10, x) & (20, y) & (40, z)
\end{array}
$$

Maximum satisfiability

$$\text{given } F = F_{hard} \wedge F_{soft} \models \bot,$$

$$\text{satisfy } F_{hard} \text{ and maximize } \sum_{c \in F_{soft}} weight(c)$$

$$\begin{array}{lclll} F_{hard} & = & (\neg x \vee \neg y) & (\neg x \vee \neg z) & (\neg y \vee \neg z) \\ F_{soft} & = & (10, x) & (20, y) & (40, z) \end{array}$$

Maximum satisfiability

$$\text{given } F = F_{hard} \wedge F_{soft} \models \bot,$$

$$\text{satisfy } F_{hard} \text{ and maximize } \sum_{c \in F_{soft}} weight(c)$$

$$\begin{array}{lcccc} F_{hard} & = & (\neg x \vee \neg y) & (\neg x \vee \neg z) & (\neg y \vee \neg z) \\ F_{soft} & = & (10, x) & (20, y) & (40, z) \end{array}$$

Maximum satisfiability

$$\text{given } F = F_{hard} \land F_{soft} \models \bot,$$

$$\text{satisfy } F_{hard} \text{ and maximize } \sum_{c \in F_{soft}} weight(c)$$

$$\begin{aligned} F_{hard} &= (\neg x \lor \neg y) \quad (\neg x \lor \neg z) \quad (\neg y \lor \neg z) \\ F_{soft} &= (10, x) \quad (20, y) \quad (40, z) \end{aligned}$$

⬇

$$\begin{aligned} F_{hard} &= (\neg x \lor \neg y) \quad (\neg x \lor \neg z) \quad (\neg y \lor \neg z) \\ F_{soft} &= (10, x) \quad (20, y) \quad (40, z) \end{aligned}$$

Scenario

Scenario

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

$|F_i|$ — up to 10^8

(e.g. Markov Logic Networks[1])

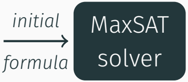
[1] R. Mangal, X. Zhang, A. Kamath, A. Nori, M. Naik: *Scaling Relational Inference Using Proofs and Refutations.* AAAI 2016: 3278–3286

Scenario

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

$|F_i|$ — up to 10^8

(e.g. Markov Logic Networks[1])



[1] R. Mangal, X. Zhang, A. Kamath, A. Nori, M. Naik: *Scaling Relational Inference Using Proofs and Refutations.* AAAI 2016: 3278–3286

Scenario

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

$|F_i|$ — up to 10^8

(e.g. Markov Logic Networks[1])

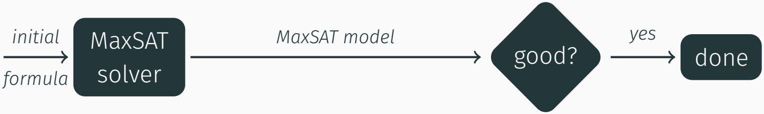
[1] R. Mangal, X. Zhang, A. Kamath, A. Nori, M. Naik: *Scaling Relational Inference Using Proofs and Refutations.* AAAI 2016: 3278–3286

Scenario

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

$|F_i|$ — up to 10^8

(e.g. Markov Logic Networks[1])



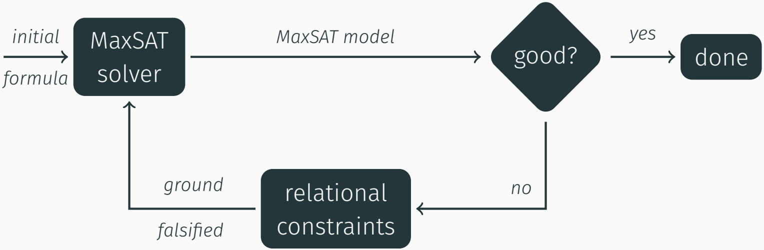
[1] R. Mangal, X. Zhang, A. Kamath, A. Nori, M. Naik: *Scaling Relational Inference Using Proofs and Refutations.* AAAI 2016: 3278–3286

Scenario

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

$|F_i|$ — up to 10^8

(e.g. Markov Logic Networks[1])



[1] R. Mangal, X. Zhang, A. Kamath, A. Nori, M. Naik: *Scaling Relational Inference Using Proofs and Refutations.* AAAI 2016: 3278–3286

Fu&Malik algorithm for MaxSAT

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$F_{soft} = (10, x) \quad (20, y) \quad (40, z)$$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$F_{soft} = (10, x) \quad (20, y) \quad (40, z)$$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$r_1 + r_2 \leqslant 1$$

$$F_{soft} = \text{~~}(10, x)\text{~~} \quad \text{~~}(20, y)\text{~~} \quad (40, z)$$

$$(10, x \vee r_1) \quad (10, y \vee r_2)$$

$$(10, y)$$

$$cost = 10$$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$r_1 + r_2 \leqslant 1$$

$$F_{soft} = \sout{(10, x)} \quad \sout{(20, y)} \quad (40, z)$$

$$(10, x \vee r_1) \quad (10, y \vee r_2)$$

$$(10, y)$$

$$cost = 10$$

Fu&Malik algorithm for MaxSAT

$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$

$r_1 + r_2 \leqslant 1 \quad r_3 + r_4 + r_5 \leqslant 1$

$F_{soft} = $ ~~$(10, x)$~~ ~~$(20, y)$~~ ~~$(40, z)$~~

~~$(10, x \vee r_1)$~~ ~~$(10, y \vee r_2)$~~

$(10, y)$

$(10, x \vee r_1 \vee r_3) \quad (10, y \vee r_2 \vee r_4) \quad (10, z \vee r_5)$

$(30, z)$

$cost = 20$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$r_1 + r_2 \leqslant 1 \quad r_3 + r_4 + r_5 \leqslant 1$$

$$F_{soft} = \sout{(10, x)} \quad \sout{(20, y)} \quad \sout{(40, z)}$$

$$\sout{(10, x \vee r_1)} \quad \sout{(10, y \vee r_2)}$$

$$(10, y)$$

$$(10, x \vee r_1 \vee r_3) \quad (10, y \vee r_2 \vee r_4) \quad (10, z \vee r_5)$$

$$(30, z)$$

$$cost = 20$$

Fu&Malik algorithm for MaxSAT

$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$

$r_1 + r_2 \leqslant 1 \quad r_3 + r_4 + r_5 \leqslant 1 \quad r_6 + r_7 \leqslant 1$

$F_{soft} =$ ~~$(10, x)$~~ ~~$(20, y)$~~ ~~$(40, z)$~~

~~$(10, x \vee r_1)$~~ ~~$(10, y \vee r_2)$~~

~~$(10, y)$~~

$(10, x \vee r_1 \vee r_3) \quad (10, y \vee r_2 \vee r_4) \quad (10, z \vee r_5)$

~~$(30, z)$~~

$(10, y \vee r_6) \quad (10, z \vee r_7)$

$(20, z)$

$cost = 30$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$r_1 + r_2 \leqslant 1 \quad r_3 + r_4 + r_5 \leqslant 1 \quad r_6 + r_7 \leqslant 1$$

$$F_{soft} = \cancel{(10, x)} \quad \cancel{(20, y)} \quad \cancel{(40, z)}$$

$$\cancel{(10, x \vee r_1)} \quad \cancel{(10, y \vee r_2)}$$

$$\cancel{(10, y)}$$

$$(10, x \vee r_1 \vee r_3) \quad (10, y \vee r_2 \vee r_4) \quad (10, z \vee r_5)$$

$$\cancel{(30, z)}$$

$$(10, y \vee r_6) \quad (10, z \vee r_7)$$

$$(20, z)$$

$$cost = 30$$

Fu&Malik algorithm for MaxSAT

$$F_{hard} = (\neg x \vee \neg y) \quad (\neg x \vee \neg z) \quad (\neg y \vee \neg z)$$

$$F_{soft} = (10, x) \quad (20, y) \quad (40, z)$$

$$cost = 30$$

Incremental approach

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \to \texttt{MaxSAT} \to F'_i$

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F_i'$
- F_i' is satisfiable

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F'_i$
- F'_i is satisfiable
- $F_{i+1} = F'_i \cup \delta_i$

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F_i'$
- F_i' is satisfiable
- $F_{i+1} = F_i' \cup \delta_i$
- MaxSAT solver **continues** working (no repetition)

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F'_i$
- F'_i is satisfiable
- $F_{i+1} = F'_i \cup \delta_i$
- MaxSAT solver **continues** working (no repetition)
- F'_i is reused at step $i + 1$

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F_i'$
- F_i' is satisfiable
- $F_{i+1} = F_i' \cup \delta_i$
- MaxSAT solver **continues** working (no repetition)
- F_i' is reused at step $i + 1$
- **two** incrementality levels:
  - SAT (keep learnt clauses)

Incremental approach

$$F_1 \subseteq F_2 \subseteq F_3 \subseteq \ldots \subseteq F_n$$

straightforward approach:

- solve each F_i independently
- $F_{i+1} = F_i \cup \delta_i$ — formulas are similar
- MaxSAT solver **repeats** its work (SAT calls included)
- inefficient

incremental approach:

- $F_i \rightarrow \texttt{MaxSAT} \rightarrow F_i'$
- F_i' is satisfiable
- $F_{i+1} = F_i' \cup \delta_i$
- MaxSAT solver **continues** working (no repetition)
- F_i' is reused at step $i + 1$
- **two** incrementality levels:
  - SAT (keep learnt clauses)
  - MaxSAT (keep unsat cores)

Poor quality cores

Poor quality cores

weighted MaxSAT splits clauses

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} = \quad (w_1, b \vee \bigvee_{i=1}^{n} a_i) \quad (w_2, \neg b) \quad \bigwedge_{i=1}^{n} (w_2, \neg a_i)$

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} = \quad (w_1, b \vee \bigvee_{i=1}^{n} a_i) \quad (w_2, \neg b) \quad \bigwedge_{i=1}^{n} (w_2, \neg a_i)$

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} =$ ~~$(w_1, b \vee \bigvee_{i=1}^n a_i)$~~ ~~$(w_2, \neg b)$~~ ~~$\bigwedge_{i=1}^n (w_2, \neg a_i)$~~

$(w_1, b \vee \bigvee_{i=1}^n a_i \vee r^1)$ $(w_1, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^n (w_1, \neg a_i \vee r_i^3)$

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} =$ ~~$(w_1, b \vee \bigvee_{i=1}^{n} a_i)$~~ ~~$(w_2, \neg b)$~~ ~~$\bigwedge_{i=1}^{n} (w_2, \neg a_i)$~~

$(w_1, b \vee \bigvee_{i=1}^{n} a_i \vee r^1)$ $(w_1, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_1, \neg a_i \vee r_i^3)$

$(w_2 - w_1, \neg b)$ $\bigwedge_{i=1}^{n} (w_2 - w_1, \neg a_i)$

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} =$ ~~$(w_1, b \vee \bigvee_{i=1}^{n} a_i)$~~ ~~$(w_2, \neg b)$~~ ~~$\bigwedge_{i=1}^{n} (w_2, \neg a_i)$~~

$(w_1, b \vee \bigvee_{i=1}^{n} a_i \vee r^1)$ $(w_1, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_1, \neg a_i \vee r_i^3)$

$(w_2 - w_1, \neg b)$ $\bigwedge_{i=1}^{n} (w_2 - w_1, \neg a_i)$

2 $F_{hard} =$ (b)

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} =$ ~~$(w_1, b \vee \bigvee_{i=1}^{n} a_i)$~~ ~~$(w_2, \neg b)$~~ ~~$\bigwedge_{i=1}^{n} (w_2, \neg a_i)$~~

$(w_1, b \vee \bigvee_{i=1}^{n} a_i \vee r^1)$ $(w_1, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_1, \neg a_i \vee r_i^3)$

$(w_2 - w_1, \neg b)$ $\bigwedge_{i=1}^{n} (w_2 - w_1, \neg a_i)$

2 $F_{hard} =$ (b)

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

$F_{soft} =$ ~~$(w_1, b \vee \bigvee_{i=1}^{n} a_i)$~~ ~~$(w_2, \neg b)$~~ ~~$\bigwedge_{i=1}^{n} (w_2, \neg a_i)$~~

$(w_1, b \vee \bigvee_{i=1}^{n} a_i \vee r^1)$ $(w_1, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_1, \neg a_i \vee r_i^3)$

~~$(w_2 - w_1, \neg b)$~~ $\bigwedge_{i=1}^{n} (w_2 - w_1, \neg a_i)$

$(w_2 - w_1, \neg b \vee r_2^2)$

$F_{hard} =$ (b)

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} = \quad (w_1, b \lor \bigvee_{i=1}^{n} a_i) \quad (w_2, \neg b) \quad \bigwedge_{i=1}^{n} (w_2, \neg a_i)$

2 $F_{hard} = \quad (b)$

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} = \quad (w_1, b \vee \bigvee_{i=1}^{n} a_i) \quad \sout{(w_2, \neg b)} \quad \bigwedge_{i=1}^{n} (w_2, \neg a_i)$

$(w_2, \neg b \vee r_1^2)$

2 $F_{hard} = \quad (b)$

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1 $F_{soft} =$ $(w_1, b \vee \bigvee_{i=1}^{n} a_i)$ ~~$(w_2, \neg b)$~~ $(w_2, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_2, \neg a_i)$

2 $F_{hard} =$ (b)

MaxSAT restarts can help!

Poor quality cores

weighted MaxSAT splits clauses

e.g. let $n \in \mathbb{N}$ and $w_1 < w_2 \in \mathbb{N}$:

1. $F_{soft} =$ $(w_1, b \vee \bigvee_{i=1}^{n} a_i)$ ~~$(w_2, \neg b)$~~ $(w_2, \neg b \vee r_1^2)$ $\bigwedge_{i=1}^{n} (w_2, \neg a_i)$
2. $F_{hard} =$ (b)

MaxSAT restarts can help!

$(split\_lim_c \leqslant k \quad \forall c \in F_{soft})$

Experimental results

Experimental evaluation

- Applications:
  1. abstraction refinement

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

     = 74 sequential MaxSAT problems

     = 669 individual MaxSAT instances (avg. 10^7 clauses)

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

  = 74 sequential MaxSAT problems

  = 669 individual MaxSAT instances (avg. 10^7 clauses)

- new approach in Open-WBO — state of the art
  1. non-incremental
  2. incremental-without-restarts
  3. incremental (clause split 2, 5, 10, 15)

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

     = 74 sequential MaxSAT problems

     = 669 individual MaxSAT instances (avg. 10^7 clauses)

- new approach in Open-WBO — state of the art
  1. non-incremental
  2. incremental-without-restarts
  3. incremental (clause split 2, 5, 10, 15)

- Machine configuration:
  - 3GHz CPU

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

  = 74 sequential MaxSAT problems

  = 669 individual MaxSAT instances (avg. 10^7 clauses)

- new approach in Open-WBO — state of the art
  1. non-incremental
  2. incremental-without-restarts
  3. incremental (clause split 2, 5, 10, 15)

- Machine configuration:
  - 3GHz CPU
  - running Linux

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

     = 74 sequential MaxSAT problems

     = 669 individual MaxSAT instances (avg. 10^7 clauses)

- new approach in Open-WBO — state of the art
  1. non-incremental
  2. incremental-without-restarts
  3. incremental (clause split 2, 5, 10, 15)

- Machine configuration:
  - 3GHz CPU
  - running Linux
  - 30m timeout

Experimental evaluation

- Applications:
  1. abstraction refinement +
  2. user-guided analysis +
  3. statistical relational inference

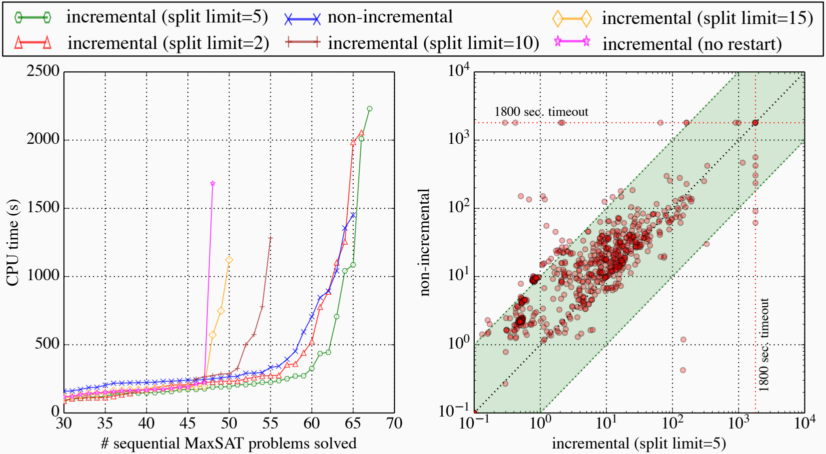
     = 74 sequential MaxSAT problems

     = 669 individual MaxSAT instances (avg. 10^7 clauses)

- new approach in Open-WBO — state of the art
  1. non-incremental
  2. incremental-without-restarts
  3. incremental (clause split 2, 5, 10, 15)

- Machine configuration:
  - 3GHz CPU
  - running Linux
  - 30m timeout
  - 32GB memout

Experimental results



(a) sequential MaxSAT problems *(b)* individual MaxSAT instances

Speedup over non-incremental approach

Split limit 5 vs. **non-incremental**:

- average speedup — 1.8×
- best speedup — 296× !

Summary and future work

Summary and future work

- new incremental approach to sequential MaxSAT:

Summary and future work

- new incremental approach to sequential MaxSAT:
    - incremental MaxSAT calls

Summary and future work

- new incremental approach to sequential MaxSAT:
  - incremental MaxSAT calls +
  - incremental SAT calls inside MaxSAT

Summary and future work

- new incremental approach to sequential MaxSAT:
  - incremental MaxSAT calls +
  - incremental SAT calls inside MaxSAT +
  - adaptive restarts

Summary and future work

- new incremental approach to sequential MaxSAT:
    - incremental MaxSAT calls +
    - incremental SAT calls inside MaxSAT +
    - adaptive restarts

- better restart strategies

Summary and future work

- new incremental approach to sequential MaxSAT:
  - incremental MaxSAT calls +
  - incremental SAT calls inside MaxSAT +
  - adaptive restarts

- better restart strategies
- state-of-the-art MaxSAT algorithms

Summary and future work

- new incremental approach to sequential MaxSAT:
  - incremental MaxSAT calls +
  - incremental SAT calls inside MaxSAT +
  - adaptive restarts

- better restart strategies
- state-of-the-art MaxSAT algorithms
- not only add but also delete clauses

Questions?