Practical SAT: from NP to ‘beyond NP’ and back

Alexey Ignatiev *(with Joao Marques-Silva and others)*

October 30, 2017

LASIGE, FC, University of Lisbon, Portugal
ISDCT SB RAS, Irkutsk, Russia

From NP to ‘beyond NP’

given a *propositional* formula in CNF,

given a *propositional* formula in CNF,
decide whether or not it is satisfiable

given a *propositional* formula in CNF,
decide whether or not it is satisfiable

Example

$\mathcal{F}_1 = (x_1) \wedge (\neg x_1 \vee \neg x_2)$

given a ***propositional*** formula in CNF,
decide whether or not it is satisfiable

Example

$\mathcal{F}_1 = (x_1) \land (\neg x_1 \lor \neg x_2)$ – satisfiable given a *propositional* formula in CNF, decide whether or not it is satisfiable

Example

$\mathcal{F}_1 = (x_1) \wedge (\neg x_1 \vee \neg x_2)$ — satisfiable

$\mathcal{F}_2 = (x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_2)$

given a *propositional* formula in CNF,
decide whether or not it is **satisfiable**

Example

$\mathcal{F}_1 = (x_1) \land (\neg x_1 \lor \neg x_2)$ – satisfiable

$\mathcal{F}_2 = (x_1) \land (\neg x_1 \lor \neg x_2) \land (x_2)$ – unsatisfiable

given a *propositional* formula in CNF,
decide whether or not it is satisfiable

Example
$\mathcal{F}_1 = (x_1) \wedge (\neg x_1 \vee \neg x_2)$ — satisfiable
$\mathcal{F}_2 = (x_1) \wedge (\neg x_1 \vee \neg x_2) \wedge (x_2)$ — unsatisfiable

first NP-complete problem!
(S. Cook. The Complexity of Theorem-Proving Procedures. *STOC*: 151–159, 1971.)

Boolean satisfiability

key to problems in NP/'beyond NP'

⬇

SAT oracles

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

Maximum satisfiability

given $\mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot$,

satisfy $\mathcal{H}$ and maximize $\sum_{c \in \mathcal{S}} weight(c)$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \textstyle\sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{rclll} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} weight(c)$$

$$\begin{array}{lclll}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
\mathcal{S} & = & (x, 10) & (y, 20) & (z, 40)
\end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \vDash \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \textstyle\sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{lcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

given $\mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot$,

satisfy $\mathcal{H}$ and maximize $\sum_{c \in \mathcal{S}} weight(c)$

$$\begin{array}{rcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{lcccc}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
\mathcal{S} & = & (x, 10) & (y, 20) & (z, 40)
\end{array}$$

$$\Downarrow$$

$$\begin{array}{lcccc}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
\mathcal{S} & = & (x, 10) & (y, 20) & (z, 40)
\end{array}$$

Simplest approach to MaxSAT

1. iterate over k

Simplest approach to MaxSAT

1. iterate over k
2. consider $\mathcal{H} \wedge \sum_{c \in \mathcal{S}} weight(c) = k$

Simplest approach to MaxSAT

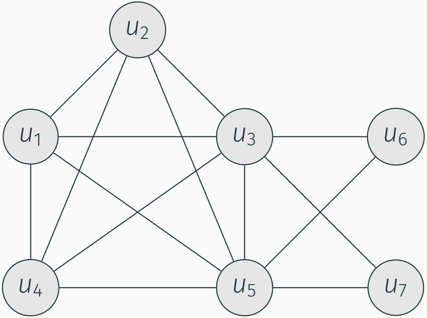
1. iterate over k

2. consider $\mathcal{H} \wedge \sum_{c \in \mathcal{S}} weight(c) = k$

3. call a **SAT oracle**

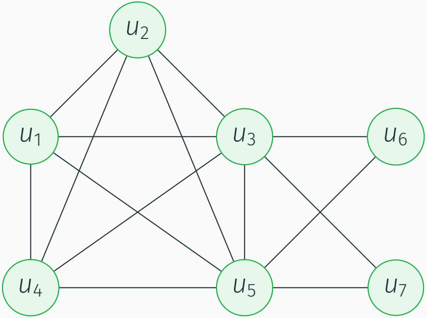
Graph optimization problems

graph $G = (V, E)$:



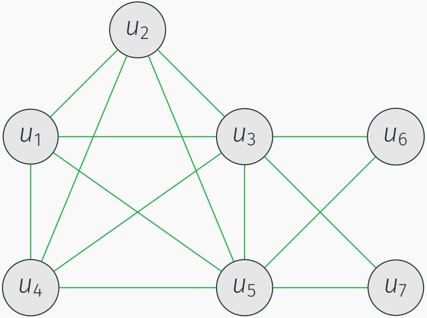
Graph optimization problems

graph $G = (V, E)$:



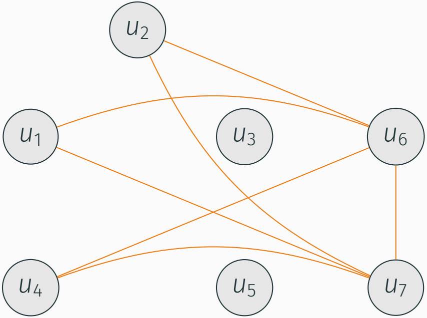
Graph optimization problems

graph $G = (V, E)$:



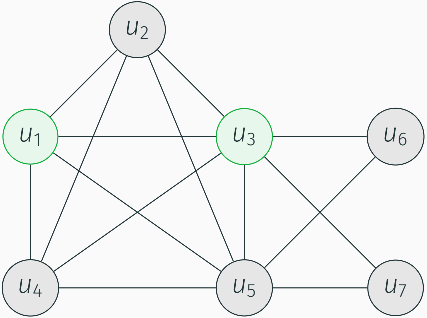
Graph optimization problems

graph $G^C = (V, E^C)$:



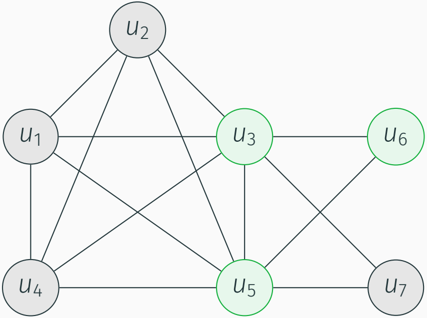
Graph optimization problems

graph $G = (V, E)$:



clique of size 2

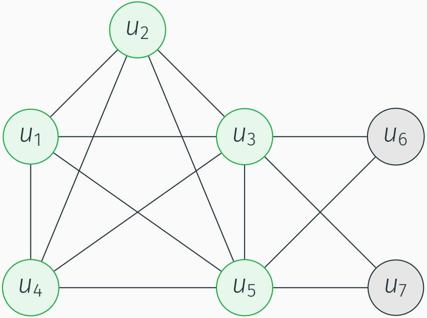
graph $G = (V, E)$:



clique of size 3

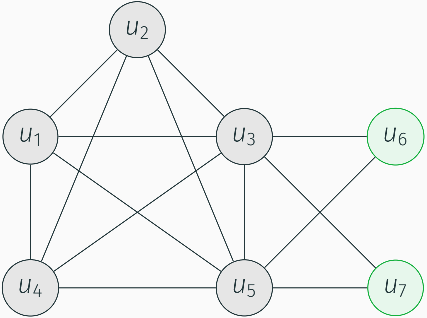
Graph optimization problems

graph $G = (V, E)$:



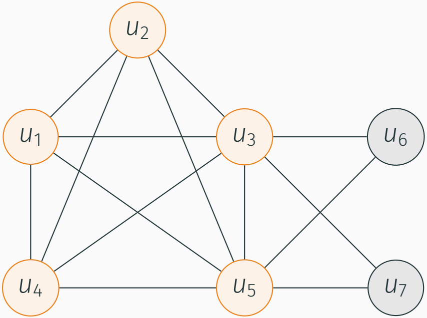
maximum clique of size 5

graph $G = (V, E)$:



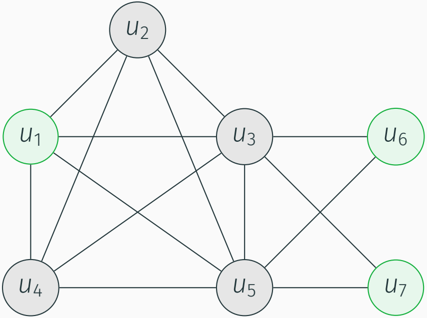
independent set of size 2

graph $G = (V, E)$:



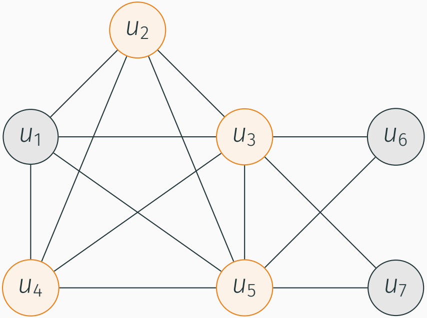
vertex cover of size 5

graph $G = (V, E)$:



maximum independent set of size 3

graph $G = (V, E)$:



minimum vertex cover of size 4

Standard encoding of maximum clique

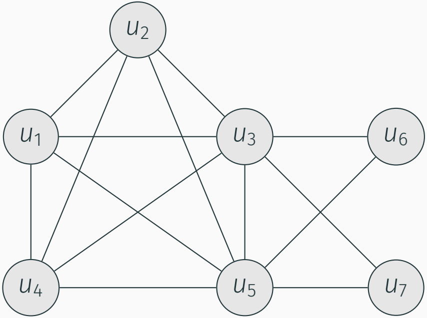
construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$

Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$

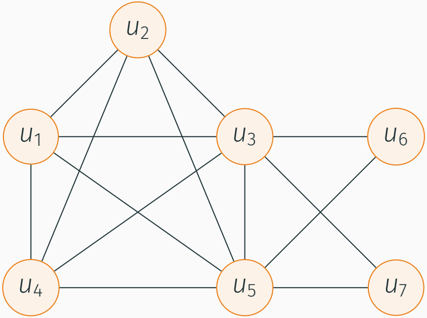
Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$



Standard encoding of maximum clique

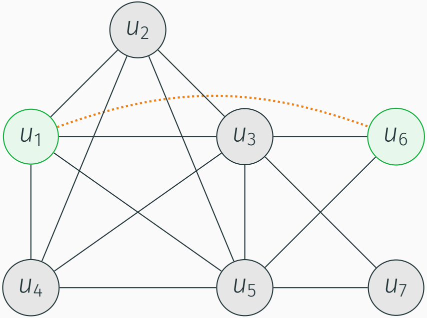
construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$



$$\mathcal{S} = \left\{ \begin{matrix} (x_1)\,(x_2)\,(x_3) \\ (x_4)\,(x_5)\,(x_6) \\ (x_7) \end{matrix} \right\}$$

Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} \triangleq \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} \triangleq \{(x_u) \mid v \in V\} \end{cases}$

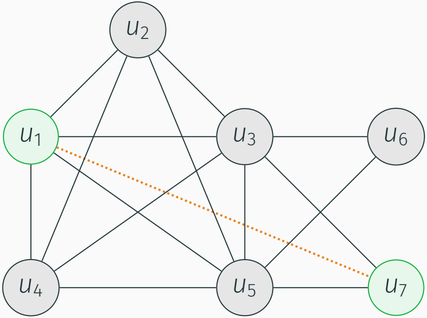


$$\mathcal{H} = \begin{Bmatrix} (\neg x_1 \vee \neg x_6) & (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6) & (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6) & (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) & \end{Bmatrix}$$

$$\mathcal{S} = \begin{Bmatrix} (x_1) \, (x_2) \, (x_3) \\ (x_4) \, (x_5) \, (x_6) \\ (x_7) \end{Bmatrix}$$

Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$

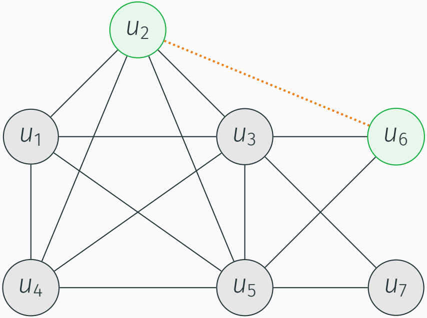


$$\mathcal{H} = \left\{ \begin{array}{l} (\neg x_1 \vee \neg x_6)\ (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6)\ (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6)\ (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) \end{array} \right\}$$

$$\mathcal{S} = \left\{ \begin{array}{l} (x_1)\ (x_2)\ (x_3) \\ (x_4)\ (x_5)\ (x_6) \\ (x_7) \end{array} \right\}$$

Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$

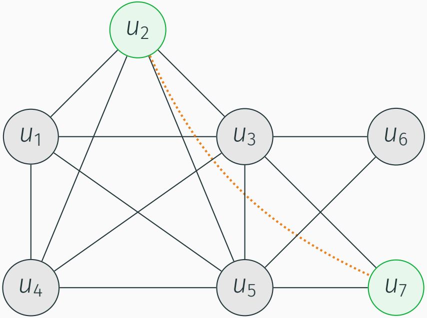


$$\mathcal{H} = \begin{Bmatrix} (\neg x_1 \vee \neg x_6) \; (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6) \; (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6) \; (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) \end{Bmatrix}$$

$$\mathcal{S} = \begin{Bmatrix} (x_1) \; (x_2) \; (x_3) \\ (x_4) \; (x_5) \; (x_6) \\ (x_7) \end{Bmatrix}$$

Standard encoding of maximum clique

$$\text{construct } \mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle \text{ s.t. } \begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$$

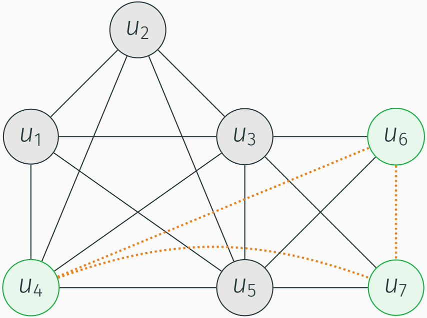


$$\mathcal{H} = \begin{Bmatrix} (\neg x_1 \vee \neg x_6) \; (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6) \; (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6) \; (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) \end{Bmatrix}$$

$$\mathcal{S} = \begin{Bmatrix} (x_1) \; (x_2) \; (x_3) \\ (x_4) \; (x_5) \; (x_6) \\ (x_7) \end{Bmatrix}$$

Standard encoding of maximum clique

$$\text{construct } \mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle \text{ s.t. } \begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u,v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$$

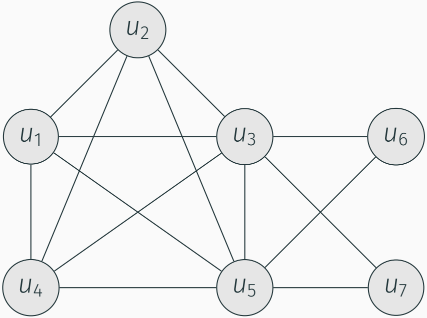


$$\mathcal{H} = \left\{ \begin{array}{l} (\neg x_1 \vee \neg x_6)\ (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6)\ (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6)\ (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) \end{array} \right\}$$

$$\mathcal{S} = \left\{ \begin{array}{l} (x_1)\ (x_2)\ (x_3) \\ (x_4)\ (x_5)\ (x_6) \\ (x_7) \end{array} \right\}$$

Standard encoding of maximum clique

construct $\mathcal{F} = \langle \mathcal{H}, \mathcal{S} \rangle$ s.t. $\begin{cases} \mathcal{H} & \triangleq & \{(\neg x_u \vee \neg x_v) \mid (u, v) \in E^C\} \\ \mathcal{S} & \triangleq & \{(x_u) \mid v \in V\} \end{cases}$



$$\mathcal{H} = \begin{Bmatrix} (\neg x_1 \vee \neg x_6) \; (\neg x_1 \vee \neg x_7) \\ (\neg x_2 \vee \neg x_6) \; (\neg x_2 \vee \neg x_7) \\ (\neg x_4 \vee \neg x_6) \; (\neg x_4 \vee \neg x_7) \\ (\neg x_6 \vee \neg x_7) \end{Bmatrix}$$

$$\mathcal{S} = \begin{Bmatrix} (x_1) \; (x_2) \; (x_3) \\ (x_4) \; (x_5) \; (x_6) \\ (x_7) \end{Bmatrix}$$

solve $\mathcal{F}$ with MaxSAT

Hidden pairwise encodings

for `ca-dblp-2012`[1],

[1] `http://networkrepository.com`

Hidden pairwise encodings

for `ca-dblp-2012`[1],

$|V| = 317\,080$ and $|E| = 1\,049\,867$

[1] `http://networkrepository.com`

Hidden pairwise encodings

for `ca-dblp-2012`[1],

$|V| = 317\,080$ and $|E| = 1\,049\,867$

⬇

$|\mathcal{H}| = |E^C| = 50\,268\,654\,793!$

[1] `http://networkrepository.com`

Hidden pairwise encodings

for `ca-dblp-2012`[1],

$|V| = 317\,080$ and $|E| = 1\,049\,867$

$\Downarrow$

$|\mathcal{H}| = |E^C| = 50\,268\,654\,793!$

$\Downarrow$

impossible to solve and **hard** to *represent*

[1] `http://networkrepository.com`

Edge cover by cliques

$$G = (V, E)$$

$T \subseteq V$ — a clique of G

($U \subseteq V$ — a clique of G^C)

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$\Rightarrow$

$|\mathcal{H}|$ grows with T^2 (or U^2)
(hidden pairwise encoding)

Edge cover by cliques

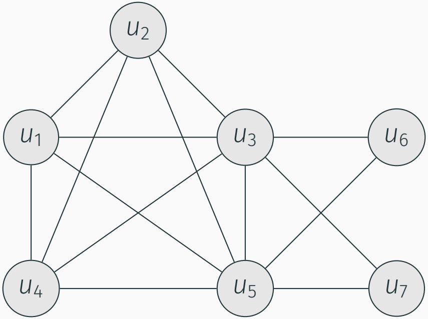
$G = (V, E)$

$T \subseteq V$ — a clique of G

($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ grows with T^2 (or U^2)

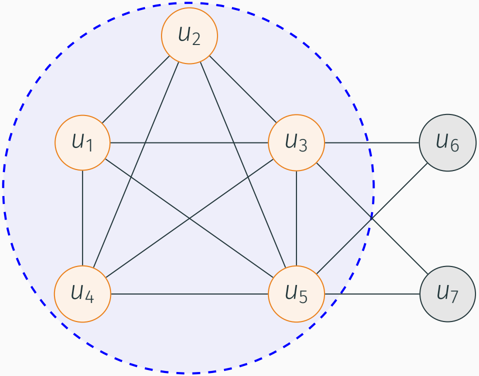
(hidden pairwise encoding)



Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ **grows with** T^2 (or U^2)
(hidden pairwise encoding)

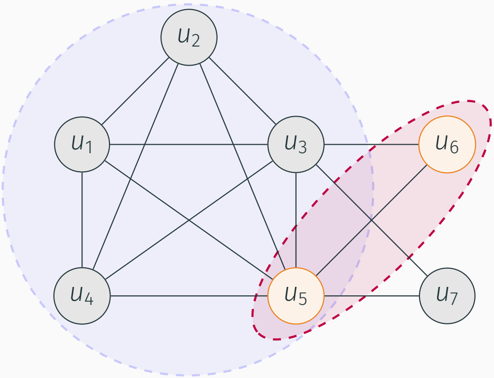


$\text{AtMost1}(x_1, x_2, x_3, x_4, x_5)$

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ grows with T^2 (or U^2)
(hidden pairwise encoding)

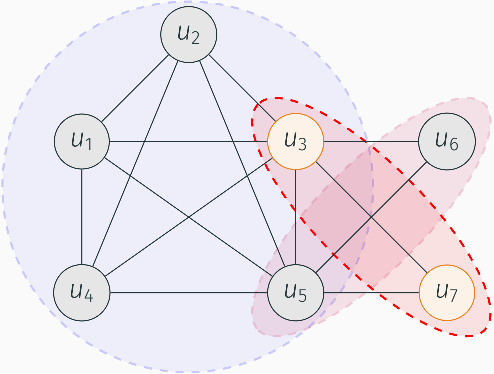


AtMost1(x_5, x_6)

AtMost1(x_1, x_2, x_3, x_4, x_5)

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ **grows with** T^2 (or U^2)
(hidden pairwise encoding)



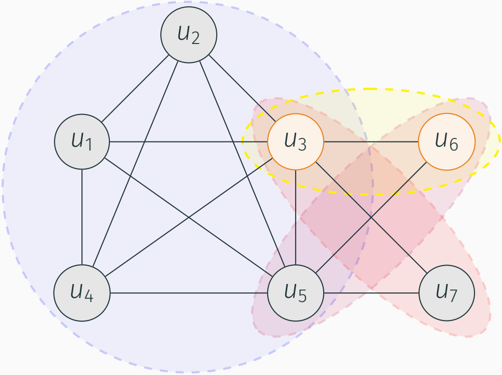
AtMost1(x_3, x_7)
AtMost1(x_5, x_6)

AtMost1(x_1, x_2, x_3, x_4, x_5)

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ **grows with** T^2 (or U^2)
(hidden pairwise encoding)



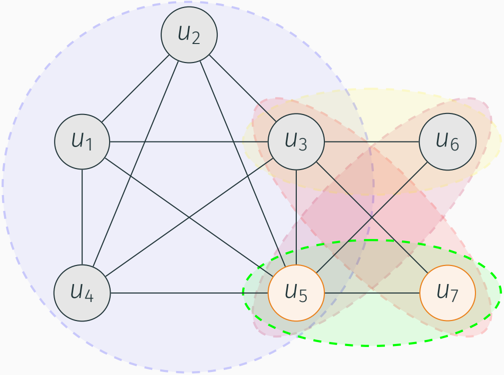
AtMost1(x_3, x_6)
AtMost1(x_3, x_7)
AtMost1(x_5, x_6)

AtMost1$(x_1, x_2, x_3, x_4, x_5)$

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ grows with T^2 (or U^2)
(hidden pairwise encoding)

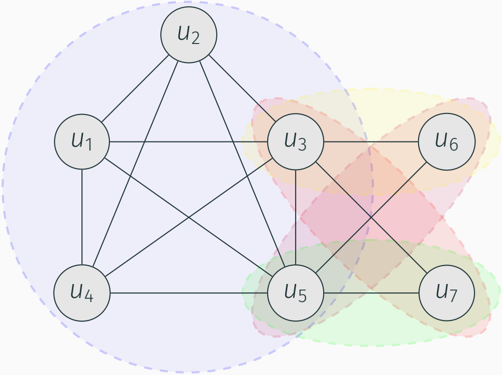


AtMost1(x_3, x_6)
AtMost1(x_3, x_7)
AtMost1(x_5, x_6)
AtMost1(x_5, x_7)
AtMost1(x_1, x_2, x_3, x_4, x_5)

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ **grows with** T^2 (or U^2)
(hidden pairwise encoding)



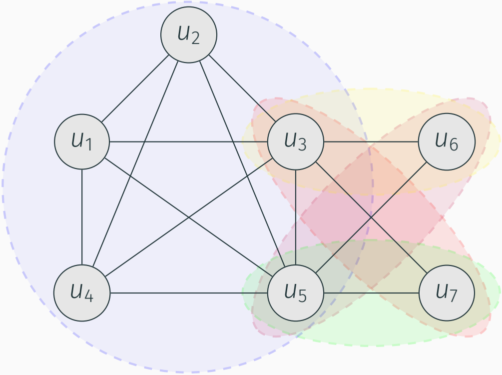
AtMost1(x_3, x_6)
AtMost1(x_3, x_7)
AtMost1(x_5, x_6)
AtMost1(x_5, x_7)
AtMost1(x_1, x_2, x_3, x_4, x_5)

pairwise encoding!

Edge cover by cliques

$G = (V, E)$
$T \subseteq V$ — a clique of G
($U \subseteq V$ — a clique of G^C)

$|\mathcal{H}|$ **grows with** T^2 (or U^2)
(hidden pairwise encoding)



$\text{AtMost1}(x_3, x_6)$
$\text{AtMost1}(x_3, x_7)$
$\text{AtMost1}(x_5, x_6)$
$\text{AtMost1}(x_5, x_7)$
$\text{AtMost1}(x_1, x_2, x_3, x_4, x_5)$

~~pairwise encoding!~~

Solving MaxClique with SAT

given $G = (V, E)$,

Solving MaxClique with SAT

given $G = (V, E)$, is there a clique of size K?

given $G = (V, E)$, is there a clique of size K?

$$\sum_{u \in V} x_u = K \tag{1}$$

given $G = (V, E)$, is there a clique of size K?

$$\sum_{u \in V} x_u = K \tag{1}$$

$$x_u \rightarrow \left(\sum_{v \in \mathrm{Adj}(u)} x_v = K - 1 \right) \tag{2}$$

filter vertices u based on ***degree*** $\mathrm{Deg}(u)$

filter vertices u based on ***degree*** $\mathrm{Deg}(u)$

$$\forall u \in V \text{ s.t. } \mathrm{Deg}(u) < K - 1: \quad (\neg x_u)$$

filter vertices u based on ***degree*** $\mathrm{Deg}(u)$

$$\forall u \in V \text{ s.t. } \mathrm{Deg}(u) < K - 1: \quad (\neg x_u)$$

$$\sum_{u \in V \wedge \mathrm{Deg}(u) \geq K-1} x_u = K$$

filter vertices u based on ***degree*** $\mathrm{Deg}(u)$

⬇

$$\forall u \in V \text{ s.t. } \mathrm{Deg}(u) < K - 1\text{:} \quad (\neg x_u)$$

$$\sum_{u \in V \wedge \mathrm{Deg}(u) \geq K-1} x_u = K$$

$$x_u \to \left(\sum_{v \in \mathrm{Adj}(u) \wedge \mathrm{Deg}(v) \geq K-1} x_v = K - 1 \right)$$

Experimental results

Experimental evaluation

- Novel SAT-based approach — SATClq

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used

Experimental evaluation

- Novel SAT-based approach — SATClq
    - implemented in Python
    - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
    - Glucose 3.0 used
    - only SAT solving time

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

1. Cliquer 1.21
2. FMC

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

1. Cliquer 1.21 2. FMC 3. IncMaxCLQ

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC

- Benchmarks:

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository 3. generated by *Benchmark*

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository 3. generated by *Benchmark*
- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository 3. generated by *Benchmark*
- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
  - running Ubuntu Linux

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository 3. generated by *Benchmark*
- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
  - running Ubuntu Linux
  - 3600s timeout

Experimental evaluation

- Novel SAT-based approach — SATClq
  - implemented in Python
  - supports MiniSat 2.2, Glucose 3.0, lingeling, etc.
  - Glucose 3.0 used
  - only SAT solving time
- Competition:

  1. Cliquer 1.21 2. FMC 3. IncMaxCLQ 4. LMC
- Benchmarks:

  1. SNAP 2. Network Repository 3. generated by *Benchmark*
- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
  - running Ubuntu Linux
  - 3600s timeout
  - 10GByte memout

Experimental evaluation

Instance	SATClq	Cliquer	FMC	IncMaxCLQ	LMC
comm-n1000	0.19	0.02	0.05	0.11	0.5
comm-n10000	—	0.99	—	12.33	0.93
ca-AstroPh	0	101.17	0.43	—	0.69
ca-citeseer	0	354.46	0.92	—	1.03
ca-coauthors-dblp	0	—	29.65	—	9.42
ca-CondMat	0	71	0.13	—	0.55
ca-dblp-2010	0	353.85	0.87	—	0.92
ca-dblp-2012	0	—	1.39	—	1.07
ca-HepPh	0	44.61	0.57	—	0.6
ca-HepTh	0	27.84	0.06	—	0.49
ca-MathSciNet	0	—	1.27	—	1.07
ia-email-EU	2.47	7.15	0.08	—	0.49
ia-reality-call	0	3.98	0.03	—	0.44
ia-retweet-pol	1.76	2.35	0.16	—	0.49
ia-wiki-Talk	—	60.48	4.21	—	0.73
rt-pol	1.7	2.39	0.19	—	0.49
rt_barackobama	0	0.46	0.45	6.63	0.46
soc-advogato	0.15	0.25	0.11	4.91	0.49
soc-epinions	—	101.67	0.21	—	0.82
soc-gplus	0.01	2.82	0.45	—	0.47
tech-as-caida2007	0.01	5.26	0.09	—	0.48
tech-internet-as	0.02	12.23	0.45	—	0.52
tech-pgp	3.05	0.71	0.07	—	0.45
tech-WHOIS	—	10.13	—	6.31	0.49
web-arabic-2005	0	151.31	2.43	—	1.57
web-baidu-baike-related	0.94	—	—	—	2.54
web-it-2004	0	—	25.32	—	4.87
web-NotreDame	0	—	3.76	—	1.37
web-sk-2005	0	97.44	0.34	—	0.64
p5sparse1	2.88	1031.15	—	12.17	0.48
p5sparse2+10clq20	1.9	24.42	—	—	0.54
p5sparse3+10clq20	3.62	150.15	—	—	0.58
p6sparse1	48.34	—	—	—	0.53
p6sparse2+10clq20	42.65	—	—	—	0.64
p6sparse3+10clq20	50.88	—	—	—	0.7
Solved (out of 35)	31	26	26	6	35

Experimental evaluation

Instance	SATClq	Cliquer	FMC	IncMaxCLQ	LMC
comm-n1000	0.19	0.02	0.05	0.11	0.5
comm-n10000	—	0.99	—	12.33	0.93
ca-AstroPh	0	101.17	0.43	—	0.69
ca-citeseer	0	354.46	0.92	—	1.03
ca-coauthors-dblp	0	—	29.65	—	9.42
ca-CondMat	0	71	0.13	—	0.55
ca-dblp-2010	0	353.85	0.87	—	0.92
ca-dblp-2012	0	—	1.39	—	1.07
ca-HepPh	0	44.61	0.57	—	0.6
ca-HepTh	0	27.84	0.06	—	0.49
ca-MathSciNet	0	—	1.27	—	1.07
ia-email-EU	2.47	7.15	0.08	—	0.49
ia-reality-call	0	3.98	0.03	—	0.44
ia-retweet-pol	1.76	2.35	0.16	—	0.49
ia-wiki-Talk	—	60.48	4.21	—	0.73
rt-pol	1.7	2.39	0.19	—	0.49
rt_barackobama	0	0.46	0.45	6.63	0.46
soc-advogato	0.15	0.25	0.11	4.91	0.49
soc-epinions	—	101.67	0.21	—	0.82
soc-gplus	0.01	2.82	0.45	—	0.47
tech-as-caida2007	0.01	5.26	0.09	—	0.48
tech-internet-as	0.02	12.23	0.45	—	0.52
tech-pgp	3.05	0.71	0.07	—	0.45
tech-WHOIS	—	10.13	—	6.31	0.49
web-arabic-2005	0	151.31	2.43	—	1.57
web-baidu-baike-related	0.94	—	—	—	2.54
web-it-2004	0	—	25.32	—	4.87
web-NotreDame	0	—	3.76	—	1.37
web-sk-2005	0	97.44	0.34	—	0.64
p5sparse1	2.88	1031.15	—	12.17	0.48
p5sparse2+10clq20	1.9	24.42	—	—	0.54
p5sparse3+10clq20	3.62	150.15	—	—	0.58
p6sparse1	48.34	—	—	—	0.53
p6sparse2+10clq20	42.65	—	—	—	0.64
p6sparse3+10clq20	50.88	—	—	—	0.7
Solved (out of 35)	**31**	**26**	**26**	**6**	**35**

Experimental evaluation

Instance	SATClq	Cliquer	FMC	IncMaxCLQ	LMC
comm-n1000	0.19	0.02	0.05	0.11	0.5
comm-n10000	—	0.99	—	12.33	0.93
ca-AstroPh	0	101.17	0.43	—	0.69
ca-citeseer	0	354.46	0.92	—	1.03
ca-coauthors-dblp	0	—	29.65	—	9.42
ca-CondMat	0	71	0.13	—	0.55
ca-dblp-2010	0	353.85	0.87	—	0.92
ca-dblp-2012	0	—	1.39	—	1.07
ca-HepPh	0	44.61	0.57	—	0.6
ca-HepTh	0	27.84	0.06	—	0.49
ca-MathSciNet	0	—	1.27	—	1.07
ia-email-EU	2.47	7.15	0.08	—	0.49
ia-reality-call	0	3.98	0.03	—	0.44
ia-retweet-pol	1.76	2.35	0.16	—	0.49
ia-wiki-Talk	—	60.48	4.21	—	0.73
rt-pol	1.7	2.39	0.19	—	0.49
rt_barackobama	0	0.46	0.45	6.63	0.46
soc-advogato	0.15	0.25	0.11	4.91	0.49
soc-epinions	—	101.67	0.21	—	0.82
soc-gplus	0.01	2.82	0.45	—	0.47
tech-as-caida2007	0.01	5.26	0.09	—	0.48
tech-internet-as	0.02	12.23	0.45	—	0.52
tech-pgp	3.05	0.71	0.07	—	0.45
tech-WHOIS	—	10.13	—	6.31	0.49
web-arabic-2005	0	151.31	2.43	—	1.57
web-baidu-baike-related	0.94	—	—	—	2.54
web-it-2004	0	—	25.32	—	4.87
web-NotreDame	0	—	3.76	—	1.37
web-sk-2005	0	97.44	0.34	—	0.64
p5sparse1	2.88	1031.15	—	12.17	0.48
p5sparse2+10clq20	1.9	24.42	—	—	0.54
p5sparse3+10clq20	3.62	150.15	—	—	0.58
p6sparse1	48.34	—	—	—	0.53
p6sparse2+10clq20	42.65	—	—	—	0.64
p6sparse3+10clq20	50.88	—	—	—	0.7
Solved (out of 35)	31	26	26	6	35

Experimental evaluation

Instance	SATClq	Cliquer	FMC	IncMaxCLQ	LMC
comm-n1000	0.19	0.02	0.05	0.11	0.5
comm-n10000	—	0.99	—	12.33	0.93
ca-AstroPh	0	101.17	0.43	—	0.69
ca-citeseer	0	354.46	0.92	—	1.03
ca-coauthors-dblp	0	—	29.65	—	9.42
ca-CondMat	0	71	0.13	—	0.55
ca-dblp-2010	0	353.85	0.87	—	0.92
ca-dblp-2012	0	—	1.39	—	1.07
ca-HepPh	0	44.61	0.57	—	0.6
ca-HepTh	0	27.84	0.06	—	0.49
ca-MathSciNet	0	—	1.27	—	1.07
ia-email-EU	2.47	7.15	0.08	—	0.49
ia-reality-call	0	3.98	0.03	—	0.44
ia-retweet-pol	1.76	2.35	0.16	—	0.49
ia-wiki-Talk	—	60.48	4.21	—	0.73
rt-pol	1.7	2.39	0.19	—	0.49
rt_barackobama	0	0.46	0.45	6.63	0.46
soc-advogato	0.15	0.25	0.11	4.91	0.49
soc-epinions	—	101.67	0.21	—	0.82
soc-gplus	0.01	2.82	0.45	—	0.47
tech-as-caida2007	0.01	5.26	0.09	—	0.48
tech-internet-as	0.02	12.23	0.45	—	0.52
tech-pgp	3.05	0.71	0.07	—	0.45
tech-WHOIS	—	10.13	—	6.31	0.49
web-arabic-2005	0	151.31	2.43	—	1.57
web-baidu-baike-related	0.94	—	—	—	2.54
web-it-2004	0	—	25.32	—	4.87
web-NotreDame	0	—	3.76	—	1.37
web-sk-2005	0	97.44	0.34	—	0.64
p5sparse1	2.88	1031.15	—	12.17	0.48
p5sparse2+10clq20	1.9	24.42	—	—	0.54
p5sparse3+10clq20	3.62	150.15	—	—	0.58
p6sparse1	48.34	—	—	—	0.53
p6sparse2+10clq20	42.65	—	—	—	0.64
p6sparse3+10clq20	50.88	—	—	—	0.7
Solved (out of 35)	31	26	26	6	35

Experimental evaluation

Instance	SATClq	Cliquer	FMC	IncMaxCLQ	LMC
comm-n1000	0.19	0.02	0.05	0.11	0.5
comm-n10000	—	0.99	—	12.33	0.93
ca-AstroPh	0	101.17	0.43	—	0.69
ca-citeseer	0	354.46	0.92	—	1.03
ca-coauthors-dblp	0	—	29.65	—	9.42
ca-CondMat	0	71	0.13	—	0.55
ca-dblp-2010	0	353.85	0.87	—	0.92
ca-dblp-2012	0	—	1.39	—	1.07
ca-HepPh	0	44.61	0.57	—	0.6
ca-HepTh	0	27.84	0.06	—	0.49
ca-MathSciNet	0	—	1.27	—	1.07
ia-email-EU	2.47	7.15	0.08	—	0.49
ia-reality-call	0	3.98	0.03	—	0.44
ia-retweet-pol	1.76	2.35	0.16	—	0.49
ia-wiki-Talk	—	60.48	4.21	—	0.73
rt-pol	1.7	2.39	0.19	—	0.49
rt_barackobama	0	0.46	0.45	6.63	0.46
soc-advogato	0.15	0.25	0.11	4.91	0.49
soc-epinions	—	101.67	0.21	—	0.82
soc-gplus	0.01	2.82	0.45	—	0.47
tech-as-caida2007	0.01	5.26	0.09	—	0.48
tech-internet-as	0.02	12.23	0.45	—	0.52
tech-pgp	3.05	0.71	0.07	—	0.45
tech-WHOIS	—	10.13	—	6.31	0.49
web-arabic-2005	0	151.31	2.43	—	1.57
web-baidu-baike-related	0.94	—	—	—	2.54
web-it-2004	0	—	25.32	—	4.87
web-NotreDame	0	—	3.76	—	1.37
web-sk-2005	0	97.44	0.34	—	0.64
p5sparse1	2.88	1031.15	—	12.17	0.48
p5sparse2+10clq20	1.9	24.42	—	—	0.54
p5sparse3+10clq20	3.62	150.15	—	—	0.58
p6sparse1	48.34	—	—	—	0.53
p6sparse2+10clq20	42.65	—	—	—	0.64
p6sparse3+10clq20	50.88	—	—	—	0.7
Solved (out of 35)	31	26	26	6	35

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)

See our papers

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)

See our papers

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)
4. Prime compilation (@ IJCAI15)

See our papers

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)
4. Prime compilation (@ IJCAI15)
5. Boolean formula minimization (@ SAT15)

See our papers

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)
4. Prime compilation (@ IJCAI15)
5. Boolean formula minimization (@ SAT15)
6. Propositional abduction (@ ECAI16)

See our papers

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)
4. Prime compilation (@ IJCAI15)
5. Boolean formula minimization (@ SAT15)
6. Propositional abduction (@ ECAI16)
7. Debugging EL+ ontologies (@ SAT15, ESWC17, DL17)

see our papers on:

1. (Q) Maximum/Minimum Satisfiability (@ SAT13, ECAI14, CJ16, . . .)
2. Package management (@ ICSE14)
3. Model-based diagnosis (@ IJCAI15)
4. Prime compilation (@ IJCAI15)
5. Boolean formula minimization (@ SAT15)
6. Propositional abduction (@ ECAI16)
7. Debugging EL+ ontologies (@ SAT15, ESWC17, DL17)
8. other topics. . . (@ IJCAI, ECAI, SAT, CP, LPAR, JELIA, EPIA, . . .)

Back to NP again

CDCL SAT solvers use resolution:

CDCL SAT solvers use resolution:

$$\frac{x \vee A \qquad\qquad \neg x \vee B}{A \vee B}$$

CDCL SAT solvers use resolution:

$$\frac{x \vee A \qquad \neg x \vee B}{A \vee B}$$

Pigeonhole principle

$m + 1$ pigeons

by m holes

Pigeonhole principle

$m+1$ pigeons by m holes $\Rightarrow$ $\exists$ hole with >1 pigeons

$m+1$ pigeons by m holes ⇒ $\exists$ hole with >1 pigeons

CNF formulation:

$x_{ij} = \mathit{true} \Leftrightarrow$ pigeon i is at hole j

Pigeonhole principle

$m+1$ pigeons by m holes ➡ $\exists$ hole with >1 pigeons

CNF formulation:

$x_{ij} = true \Leftrightarrow$ pigeon i is at hole j

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

$m+1$ pigeons by m holes ➡ $\exists$ hole with >1 pigeons

CNF formulation:

$x_{ij} = \mathit{true} \Leftrightarrow$ pigeon i is at hole j

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

⇩

hard for resolution!

(A. Haken. The intractability of resolution. *TCS*, 39:297–308, 1985.)

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \perp,$$

given $\mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot$,

satisfy $\mathcal{H}$ and maximize $\sum_{c \in \mathcal{S}} \mathit{weight}(c)$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{rcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} weight(c)$$

$$\begin{array}{rcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \textstyle\sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{lcccc}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
\mathcal{S} & = & (x, 10) & (y, 20) & (z, 40)
\end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} \mathit{weight}(c)$$

$$\begin{array}{lcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

Maximum satisfiability

$$\text{given } \mathcal{F} = \mathcal{H} \wedge \mathcal{S} \models \bot,$$

$$\text{satisfy } \mathcal{H} \text{ and maximize } \sum_{c \in \mathcal{S}} weight(c)$$

$$\begin{array}{lcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

$$\Downarrow$$

$$\begin{array}{lcccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 10) & (y, 20) & (z, 40) \end{array}$$

MSU3 algorithm for MaxSAT

$$\begin{array}{ccccc} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ \mathcal{S} & = & (x, 1) & (y, 1) & (z, 1) \end{array}$$

MSU3 algorithm for MaxSAT

$$\mathcal{H} = (\neg x \vee \neg y, \top) \quad (\neg x \vee \neg z, \top) \quad (\neg y \vee \neg z, \top)$$

$$\mathcal{S} = (x, 1) \quad (y, 1) \quad (z, 1)$$

MSU3 algorithm for MaxSAT

$$
\begin{array}{lllll}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
 & & (r_1 + r_2 \leq 1, \top) & & \\
\mathcal{S} & = & \cancel{(x, 1)} & \cancel{(y, 1)} & (z, 1) \\
 & & (x \vee r_1, 1) & (y \vee r_2, 1) &
\end{array}
$$

MSU3 algorithm for MaxSAT

$$\begin{array}{lllll} \mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\ & & (r_1 + r_2 \leq 1, \top) & & \\ \mathcal{S} & = & \cancel{(x, 1)} & \cancel{(y, 1)} & (z, 1) \\ & & (x \vee r_1, 1) & (y \vee r_2, 1) & \end{array}$$

MSU3 algorithm for MaxSAT

$$
\begin{array}{lclll}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
 & & \cancel{(r_1 + r_2 \leq 1, \top)} & (r_1 + r_2 + r_3 \leq 2, \top) & \\
\mathcal{S} & = & \cancel{(x, 1)} & \cancel{(y, 1)} & \cancel{(z, 1)} \\
 & & (x \vee r_1, 1) & (y \vee r_2, 1) & \\
 & & & & (z \vee r_3, 1)
\end{array}
$$

MSU3 algorithm for MaxSAT

$$
\begin{array}{lcccc}
\mathcal{H} & = & (\neg x \vee \neg y, \top) & (\neg x \vee \neg z, \top) & (\neg y \vee \neg z, \top) \\
 & & \cancel{(r_1 + r_2 \leq 1, \top)} & (r_1 + r_2 + r_3 \leq 2, \top) & \\
\mathcal{S} & = & \cancel{(x, 1)} & \cancel{(y, 1)} & \cancel{(z, 1)} \\
 & & (x \vee r_1, 1) & (y \vee r_2, 1) & \\
 & & & & (z \vee r_3, 1)
\end{array}
$$

MSU3 algorithm for MaxSAT

$$\mathcal{H} = (\neg x \vee \neg y, \top) \quad (\neg x \vee \neg z, \top) \quad (\neg y \vee \neg z, \top)$$

$$\cancel{(r_1 + r_2 \leq 1, \top)} \quad (r_1 + r_2 + r_3 \leq 2, \top)$$

$$\mathcal{S} = \cancel{(x, 1)} \quad \cancel{(y, 1)} \quad \cancel{(z, 1)}$$

$$(x \vee r_1, 1) \quad (y \vee r_2, 1)$$

$$(z \vee r_3, 1)$$

$$cost = 2$$

Solving SAT with Horn MaxSAT

Approach

input: $\mathcal{F}$

$\texttt{HEnc}(\mathcal{F}) = \langle \mathcal{H}, \mathcal{S} \rangle \leftarrow \texttt{DualRailEncode}(\mathcal{F})$

$\text{cost} \leftarrow \texttt{ApplyMaxSAT}(\texttt{HEnc}(\mathcal{F}))$

Approach

input: $\mathcal{F}$

$\texttt{HEnc}(\mathcal{F}) = \langle \mathcal{H}, \mathcal{S} \rangle \leftarrow \texttt{DualRailEncode}(\mathcal{F})$

cost $\leftarrow \texttt{ApplyMaxSAT}(\texttt{HEnc}(\mathcal{F}))$

if cost $\leq |\mathrm{var}(\mathcal{F})|$**:**

    return *true*

else:

    return *false*

Dual-rail encoding

$\forall x_i \in \mathrm{var}(\mathcal{F})$

Dual-rail encoding

$$\forall x_i \in \mathrm{var}(\mathcal{F}) \quad \Rightarrow \quad \begin{cases} (p_i, 1) \\ (n_i, 1) \\ (\neg p_i \vee \neg n_i, \top) \end{cases}$$

Dual-rail encoding

$$\forall x_i \in \mathrm{var}(\mathcal{F}) \quad \Rightarrow \quad \begin{cases} (p_i, 1) \\ (n_i, 1) \\ (\neg p_i \vee \neg n_i, \top) \end{cases}$$

$$\forall c_i \in \mathcal{F},$$
$$c_i = (l_{i1} \vee \ldots \vee l_{ik_i})$$

Dual-rail encoding

$$\forall x_i \in \mathrm{var}(\mathcal{F}) \quad \Rightarrow \quad \begin{cases} (p_i, 1) \\ (n_i, 1) \\ (\neg p_i \vee \neg n_i, \top) \end{cases}$$

$$\begin{gathered}\forall c_i \in \mathcal{F}, \\ c_i = (l_{i1} \vee \ldots \vee l_{ik_i})\end{gathered} \quad \Rightarrow \quad \begin{cases} (\neg y_{i1} \vee \ldots \vee \neg y_{ik_i}, \top) \text{ s.t.} \\ y_{ij} \leftarrow p_{ij} \text{ if } l_{ij} = \neg x_{ij} \\ y_{ij} \leftarrow n_{ij} \text{ if } l_{ij} = x_{ij} \end{cases}$$

Dual-rail encoding

$$\forall x_i \in \mathrm{var}(\mathcal{F}) \quad \Rightarrow \quad \begin{cases} (p_i, 1) \\ (n_i, 1) \\ (\neg p_i \vee \neg n_i, \top) \end{cases}$$

$$\begin{array}{c} \forall c_i \in \mathcal{F}, \\ c_i = (l_{i1} \vee \ldots \vee l_{ik_i}) \end{array} \quad \Rightarrow \quad \begin{cases} (\neg y_{i1} \vee \ldots \vee \neg y_{ik_i}, \top) \text{ s.t.} \\ y_{ij} \leftarrow p_{ij} \text{ if } l_{ij} = \neg x_{ij} \\ y_{ij} \leftarrow n_{ij} \text{ if } l_{ij} = x_{ij} \end{cases}$$

$$\Downarrow$$

Horn MaxSAT formula!

Dual-rail encoding (example)

$$\mathcal{F} \qquad\qquad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

Dual-rail encoding (example)

$$\mathcal{F} \qquad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$$\mathcal{S} \qquad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$$

$$\mathcal{P} \qquad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$$

Dual-rail encoding (example)

$$\mathcal{F} \qquad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$$\mathcal{S} \qquad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$$

$$\mathcal{P} \qquad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$$

$$\mathcal{H} \qquad (\neg p_1 \vee \neg p_2, \top) \quad (\neg n_2, \top)$$

Dual-rail encoding (example)

$\mathcal{F}$ $\quad (\neg x_1 \vee \neg x_2) \quad (x_2)$

$\Downarrow$

$\mathcal{S}$ $\quad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$

$\mathcal{P}$ $\quad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$

$\mathcal{H}$ $\quad (\neg p_1 \vee \neg p_2, \top) \quad (\neg n_2, \top)$

Dual-rail encoding (example)

$$\mathcal{F} \qquad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$$\mathcal{S} \qquad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$$

$$\mathcal{P} \qquad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$$

$$\mathcal{H} \qquad (\neg p_1 \vee \neg p_2, \top) \qquad (\neg n_2, \top)$$

Dual-rail encoding (example)

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

$\mathcal{S}$ $(p_1, 1)$ $(n_1, 1)$ $(p_2, 1)$ $(n_2, 1)$

$\mathcal{P}$ $(\neg p_1 \vee \neg n_1, \top)$ $(\neg p_2 \vee \neg n_2, \top)$

$\mathcal{H}$ $(\neg p_1 \vee \neg p_2, \top)$ $(\neg n_2, \top)$

$\text{cost} = 2$

($\mathcal{F}$ is satisfiable)

Dual-rail encoding (example)

$$\mathcal{F} \quad (x_1) \quad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$$\mathcal{S} \quad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$$

$$\mathcal{P} \quad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$$

$$\mathcal{H} \quad (\neg p_1 \vee \neg p_2, \top) \quad (\neg n_2, \top)$$

Dual-rail encoding (example)

$$\mathcal{F} \quad (x_1) \quad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$$\mathcal{S} \quad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$$

$$\mathcal{P} \quad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$$

$$\mathcal{H} \quad (\neg n_1, \top) \quad (\neg p_1 \vee \neg p_2, \top) \quad (\neg n_2, \top)$$

Dual-rail encoding (example)

$$\mathcal{F} \quad (x_1) \quad (\neg x_1 \vee \neg x_2) \quad (x_2)$$

$$\Downarrow$$

$\mathcal{S}$ $\quad (p_1, 1) \quad (n_1, 1) \quad (p_2, 1) \quad (n_2, 1)$

$\mathcal{P}$ $\quad (\neg p_1 \vee \neg n_1, \top) \quad (\neg p_2 \vee \neg n_2, \top)$

$\mathcal{H}$ $\quad (\neg n_1, \top) \quad (\neg p_1 \vee \neg p_2, \top) \quad (\neg n_2, \top)$

$\text{cost} = 3$

($\mathcal{F}$ is unsatisfiable)

Dual-rail encoding PHP

$x_{ij}, \quad \left.\begin{matrix} 1 \le i \le m+1 \\ 1 \le j \le m \end{matrix}\right\} m \cdot (m+1)$ vars $\Rightarrow$ n_{ij} and p_{ij}

Dual-rail encoding PHP

$x_{ij},\quad \left.\begin{array}{c} 1 \leq i \leq m+1 \\ 1 \leq j \leq m \end{array}\right\} m \cdot (m+1)$ vars $\Rightarrow$ n_{ij} and p_{ij}

$\mathcal{P}$ $\quad \{(\neg p_{ij} \vee \neg n_{ij}, \top) \mid 1 \leq i \leq m+1, 1 \leq j \leq m\}$

Dual-rail encoding PHP

$x_{ij}, \quad \left.\begin{array}{c} 1 \leq i \leq m+1 \\ 1 \leq j \leq m \end{array}\right\} m \cdot (m+1) \text{ vars}$ ➡ n_{ij} and p_{ij}

$\mathcal{P}$ $\{(\neg p_{ij} \vee \neg n_{ij}, \top) \mid 1 \leq i \leq m+1, 1 \leq j \leq m\}$

$\mathcal{L}_i$ $\text{AtLeast1}(x_{i1}, \ldots, x_{im}) = (x_{i1} \vee \ldots \vee x_{im})$ ➡ $(\neg n_{i1} \vee \ldots \vee \neg n_{im}, \top)$

Dual-rail encoding PHP

$$x_{ij}, \quad \left.\begin{array}{c} 1 \le i \le m+1 \\ 1 \le j \le m \end{array}\right\} m \cdot (m+1) \text{ vars} \qquad n_{ij} \text{ and } p_{ij}$$

$$\mathcal{P} \qquad \{(\neg p_{ij} \vee \neg n_{ij}, \top) \mid 1 \le i \le m+1, 1 \le j \le m\}$$

$$\mathcal{L}_i \qquad \text{AtLeast1}(x_{i1}, \ldots, x_{im}) = (x_{i1} \vee \ldots \vee x_{im}) \qquad (\neg n_{i1} \vee \ldots \vee \neg n_{im}, \top)$$

$$\mathcal{M}_j \qquad \text{AtMost1}(x_{1j}, \ldots, x_{m+1,j}) = \{(\neg x_{kj} \vee \neg x_{lj}) \mid 1 \le k < l \le m+1\} \qquad \{(\neg p_{kj} \vee \neg p_{lj}, \top) \mid 1 \le k < l \le m+1\}$$

Dual-rail encoding PHP

x_{ij}, $\left.\begin{matrix} 1 \le i \le m+1 \\ 1 \le j \le m \end{matrix}\right\} m \cdot (m+1)$ vars $\Rightarrow$ n_{ij} and p_{ij}

$\mathcal{P}$ $\{(\neg p_{ij} \vee \neg n_{ij}, \top) \mid 1 \le i \le m+1, 1 \le j \le m\}$

$\mathcal{L}_i$ $\text{AtLeast1}(x_{i1}, \dots, x_{im}) = (x_{i1} \vee \dots \vee x_{im})$ $\Rightarrow$ $(\neg n_{i1} \vee \dots \vee \neg n_{im}, \top)$

$\mathcal{M}_j$ $\text{AtMost1}(x_{1j}, \dots, x_{m+1,j}) = \{(\neg x_{kj} \vee \neg x_{lj}) \mid 1 \le k < l \le m+1\}$ $\Rightarrow$ $\{(\neg p_{kj} \vee \neg p_{lj}, \top) \mid 1 \le k < l \le m+1\}$

$\mathcal{S}$ $\{(p_{ij}, 1), (n_{ij}, 1) \mid 1 \le i \le m+1, 1 \le j \le m\}$

Dual-rail encoding PHP

$x_{ij}, \quad \left.\begin{matrix} 1 \le i \le m+1 \\ 1 \le j \le m \end{matrix}\right\} m \cdot (m+1) \text{ vars} \quad \Rightarrow \quad n_{ij} \text{ and } p_{ij}$

$\mathcal{P} \quad \{(\neg p_{ij} \vee \neg n_{ij}, \top) \mid 1 \le i \le m+1, 1 \le j \le m\}$

$\mathcal{L}_i \quad \begin{matrix} \text{AtLeast1}(x_{i1}, \ldots, x_{im}) = \\ (x_{i1} \vee \ldots \vee x_{im}) \end{matrix} \quad \Rightarrow \quad (\neg n_{i1} \vee \ldots \vee \neg n_{im}, \top)$

$\mathcal{M}_j \quad \begin{matrix} \text{AtMost1}(x_{1j}, \ldots, x_{m+1,j}) = \\ \{(\neg x_{kj} \vee \neg x_{lj}) \mid 1 \le k < l \le m+1\} \end{matrix} \quad \Rightarrow \quad \begin{matrix} \{(\neg p_{kj} \vee \neg p_{lj}, \top) \mid \\ 1 \le k < l \le m+1\} \end{matrix}$

$\mathcal{S} \quad \{(p_{ij}, 1), (n_{ij}, 1) \mid 1 \le i \le m+1, 1 \le j \le m\}$

$$\text{HEnc}(\text{PHP}_m^{m+1}) \triangleq \langle \mathcal{H}, \mathcal{S} \rangle = \left\langle \bigwedge_{i=1}^{m+1} \mathcal{L}_i \wedge \bigwedge_{j=1}^{m} \mathcal{M}_j \wedge \mathcal{P}, \mathcal{S} \right\rangle$$

DRE+MaxSAT for PHP in polynomial time

1. assume MSU3 algorithm
    - analyze ***disjoint*** sets separately

DRE+MaxSAT for PHP in polynomial time

1. assume MSU3 algorithm
   - analyze ***disjoint*** sets separately
2. relate soft clauses with each $\mathcal{L}_i$ and $\mathcal{M}_j$
   - each constraint disjoint from the others — but not from $\mathcal{P}$

DRE+MaxSAT for PHP in polynomial time

1. assume MSU3 algorithm
   - analyze ***disjoint*** sets separately
2. relate soft clauses with each $\mathcal{L}_i$ and $\mathcal{M}_j$
   - each constraint disjoint from the others — but not from $\mathcal{P}$
3. derive large enough lower bound on # of falsified clauses:

DRE+MaxSAT for PHP in polynomial time

1. assume MSU3 algorithm
   - analyze ***disjoint*** sets separately

2. relate soft clauses with each $\mathcal{L}_i$ and $\mathcal{M}_j$
   - 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	$j = 1, \ldots, m$	$m \cdot m$
			$m \cdot (m+1) + 1$

DRE+MaxSAT for PHP in polynomial time

1. assume MSU3 algorithm
   - analyze ***disjoint*** sets separately
2. relate soft clauses with each $\mathcal{L}_i$ and $\mathcal{M}_j$
   - 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	$j = 1, \ldots, m$	$m \cdot m$
			$m \cdot (m+1) + 1$

4. each lower bound increase — by unit propagation

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	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
	$(\neg p_{1j} \vee \neg p_{3j})$, $(\neg p_{2j} \vee \neg p_{3j})$, $(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$, $\sum_{l=1}^{2} r_{lj} \leq 1$	(p_{3j})	$(r_{3j} \vee p_{3j})$	$\sum_{l=1}^{3} r_{lj} \leq 2$	1
	$\cdots$ $(m-3$ times$)$				
	$(\neg p_{1j} \vee \neg p_{m+1j}), \ldots,$ $(\neg p_{mj} \vee \neg p_{m+1j})$, $(r_{1j} \vee p_{1j}), \ldots,$ $(r_{mj} \vee p_{mj})$, $\sum_{l=1}^{m} r_{lj} \leq m-1$	(p_{m+1j})	$(r_{m+1j} \vee p_{m+1j})$	$\sum_{l=1}^{m+1} r_{lj} \leq m$	1

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	1

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	1
	$(\neg p_{1j} \vee \neg p_{2j})$	$(p_{1j}), (p_{2j})$	$(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$	$\sum_{l=1}^{2} r_{lj} \leq 1$	1
$\mathcal{M}_j$					

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	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
	$(\neg p_{1j} \vee \neg p_{3j})$, $(\neg p_{2j} \vee \neg p_{3j})$, $(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$, $\sum_{l=1}^{2} r_{lj} \leq 1$	(p_{3j})	$(r_{3j} \vee p_{3j})$	$\sum_{l=1}^{3} r_{lj} \leq 2$	1

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	1
	$(\neg p_{1j} \vee \neg p_{2j})$	$(p_{1j}), (p_{2j})$	$(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$	$\sum_{l=1}^{2} r_{lj} \leq 1$	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

$\cdots$ ($m - 3$ times)

DRE+MaxSAT for PHP in polynomial time

Constraint	Hard clause(s)	Soft clause(s)	Relaxed clauses	Updated AtMost$k$ Constraints	LB increase
$\mathcal{L}_i$	$(\neg n_{i1} \vee \ldots \vee \neg n_{im})$	$(n_{i1}), \ldots, (n_{im})$	$(r_{il} \vee n_{il})$, $1 \leq l \leq m$	$\sum_{l=1}^{m} r_{il} \leq 1$	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
	$(\neg p_{1j} \vee \neg p_{3j})$, $(\neg p_{2j} \vee \neg p_{3j})$, $(r_{1j} \vee p_{1j})$, $(r_{2j} \vee p_{2j})$, $\sum_{l=1}^{2} r_{lj} \leq 1$	(p_{3j})	$(r_{3j} \vee p_{3j})$	$\sum_{l=1}^{3} r_{lj} \leq 2$	1
			$\cdots$ ($m - 3$ times)		
	$(\neg p_{1j} \vee \neg p_{m+1j}), \ldots$, $(\neg p_{mj} \vee \neg p_{m+1j})$, $(r_{1j} \vee p_{1j}), \ldots$, $(r_{mj} \vee p_{mj})$, $\sum_{l=1}^{m} r_{lj} \leq m - 1$	(p_{m+1j})	$(r_{m+1j} \vee p_{m+1j})$	$\sum_{l=1}^{m+1} r_{lj} \leq m$	1

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

- **Key points**:
  - for each $\mathcal{L}_i$, UP raises LB by 1

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

- **Key points**:
    - for each $\mathcal{L}_i$, UP raises LB by 1
    - for each $\mathcal{M}_j$, UP raises LB by m

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

- **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 $m \cdot (m + 1) + 1$

DRE+MaxSAT for PHP — unit propagation steps in $\mathcal{M}_j$

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

- **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 $m \cdot (m+1) + 1$
  - PHP_m^{m+1} is ***unsatisfiable***

Short MaxSAT proof for PHP

short DRE+MaxSAT-resolution proof

⬇

see the paper!

(Ignatiev et al. @ SAT 2017)

Experimental results

Experimental evaluation

- approaches tested:
  1. **SAT** (MiniSat 2.2, *Glucose 3.0*)
  2. **SAT+** (*lingeling*, CryptoMiniSat)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)
  4. CG MaxSAT (*MSCG*, *OpenWBO16*, WPM3)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)
  4. CG MaxSAT (*MSCG*, *OpenWBO16*, WPM3)
  5. MaxRes (*Eva*)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)
  4. CG MaxSAT (*MSCG*, *OpenWBO16*, WPM3)
  5. MaxRes (*Eva*)
  6. MIP (*CPLEX*)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)
  4. CG MaxSAT (*MSCG*, *OpenWBO16*, WPM3)
  5. MaxRes (*Eva*)
  6. MIP (*CPLEX*)
  7. OPB (*cdcl-cuttingplanes*, Sat4j)

Experimental evaluation

- approaches tested:
    1. **SAT** (MiniSat 2.2, *Glucose 3.0*)
    2. **SAT+** (*lingeling*, CryptoMiniSat)
    3. **IHS MaxSAT** (*MaxHS*, *LMHS*)
    4. **CG MaxSAT** (*MSCG*, *OpenWBO16*, WPM3)
    5. **MaxRes** (*Eva*)
    6. **MIP** (*CPLEX*)
    7. **OPB** (*cdcl-cuttingplanes*, Sat4j)
    8. **BDD** (*ZRes*)

Experimental evaluation

- approaches tested:
  1. SAT (MiniSat 2.2, *Glucose 3.0*)
  2. SAT+ (*lingeling*, CryptoMiniSat)
  3. IHS MaxSAT (*MaxHS*, *LMHS*)
  4. CG MaxSAT (*MSCG*, *OpenWBO16*, WPM3)
  5. MaxRes (*Eva*)
  6. MIP (*CPLEX*)
  7. OPB (*cdcl-cuttingplanes*, Sat4j)
  8. BDD (*ZRes*)

- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM

Experimental evaluation

- approaches tested:
  1. **SAT** (MiniSat 2.2, *Glucose 3.0*)
  2. **SAT+** (*lingeling*, CryptoMiniSat)
  3. **IHS MaxSAT** (*MaxHS*, *LMHS*)
  4. **CG MaxSAT** (*MSCG*, *OpenWBO16*, WPM3)
  5. **MaxRes** (*Eva*)
  6. **MIP** (*CPLEX*)
  7. **OPB** (*cdcl-cuttingplanes*, Sat4j)
  8. **BDD** (*ZRes*)

- Machine configuration:
  - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
  - running Ubuntu Linux

Experimental evaluation

- approaches tested:
    1. **SAT** (MiniSat 2.2, *Glucose 3.0*)
    2. **SAT+** (*lingeling*, CryptoMiniSat)
    3. **IHS MaxSAT** (*MaxHS*, *LMHS*)
    4. **CG MaxSAT** (*MSCG*, *OpenWBO16*, WPM3)
    5. **MaxRes** (*Eva*)
    6. **MIP** (*CPLEX*)
    7. **OPB** (*cdcl-cuttingplanes*, Sat4j)
    8. **BDD** (*ZRes*)

- Machine configuration:
    - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
    - running Ubuntu Linux
    - 1800s timeout

Experimental evaluation

- approaches tested:
    1. **SAT** (MiniSat 2.2, *Glucose 3.0*)
    2. **SAT+** (*lingeling*, CryptoMiniSat)
    3. **IHS MaxSAT** (*MaxHS*, *LMHS*)
    4. **CG MaxSAT** (*MSCG*, *OpenWBO16*, WPM3)
    5. **MaxRes** (*Eva*)
    6. **MIP** (*CPLEX*)
    7. **OPB** (*cdcl-cuttingplanes*, Sat4j)
    8. **BDD** (*ZRes*)

- Machine configuration:
    - Intel Xeon E5-2630 2.60GHz with 64GByte RAM
    - running Ubuntu Linux
    - 1800s timeout
    - 10GByte memout

Experimental evaluation

- benchmarks:
  1. PHP (pigeonhole principle):
     - PHP_m^{m+1}, $m \in \{4, \ldots, 100\}$
     - pairwise — 46 instances
     - sequential counter — 46 instances

[2] P. Chatalic and L. Simon. Multiresolution for SAT checking. *International Journal on Artificial Intelligence Tools*, 10(4):451–481, 2001.

Experimental evaluation

- benchmarks:
  1. PHP (pigeonhole principle):
     - PHP_m^{m+1}, $m \in \{4, \ldots, 100\}$
     - pairwise — 46 instances
     - sequential counter — 46 instances
  2. URQ (Urquhart formulas[2]):
     - $\mathrm{URQ}_{n,i}$, $n \in \{3, \ldots, 30\}, i \in \{1, 2, 3\}$
     - 84 instances

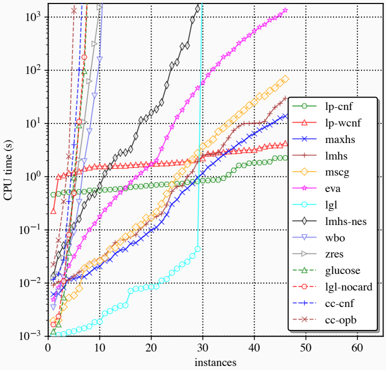
[2] P. Chatalic and L. Simon. Multiresolution for SAT checking. *International Journal on Artificial Intelligence Tools*, 10(4):451–481, 2001.

Experimental evaluation

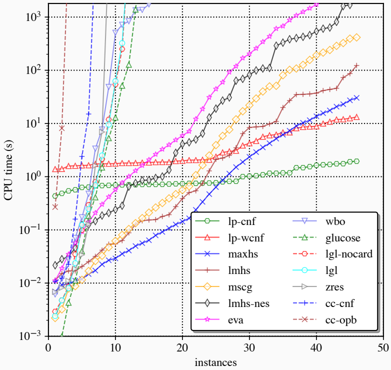
- benchmarks:
  1. PHP (pigeonhole principle):
     - PHP_m^{m+1} , $m \in \{4, \ldots, 100\}$
     - pairwise — 46 instances
     - sequential counter — 46 instances
  2. URQ (Urquhart formulas[2]):
     - $\mathrm{URQ}_{n,i}$, $n \in \{3, \ldots, 30\}, i \in \{1, 2, 3\}$
     - 84 instances
  3. COMB (combined):
     - $\mathrm{PHP}_m^{m+1} \vee \mathrm{URQ}_{n,i}$, $m \in \{7, 9, 11, 13\}, n \in \{3, \ldots, 10\}, i \in \{1, 2, 3\}$
     - 96 instances

[2] P. Chatalic and L. Simon. Multiresolution for SAT checking. *International Journal on Artificial Intelligence Tools*, 10(4):451–481, 2001.

Performance on pigeonhole formulas



(a) PHP-pw *(pairwise)*



(b) PHP-sc *(sequential counter)*

$\mathcal{P}$ clauses can be harmful

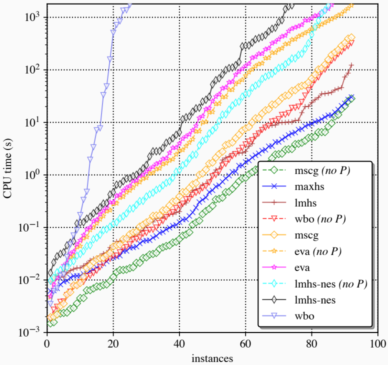
$(\neg p_i \vee \neg n_i, \top)$

$\mathcal{P}$ clauses can be harmful

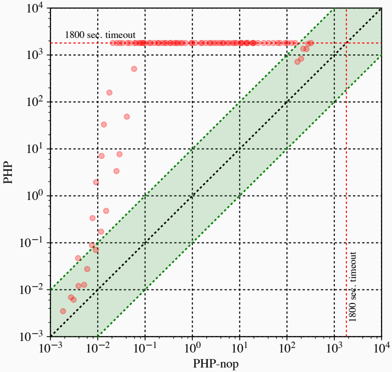
$(\neg p_i \vee \neg n_i, \top) \wedge (p_i, 1) \wedge (n_i, 1)$ – trivial core

$\mathcal{P}$ clauses can be harmful

$(\neg p_i \vee \neg n_i, \top) \wedge (p_i, 1) \wedge (n_i, 1)$ — trivial core

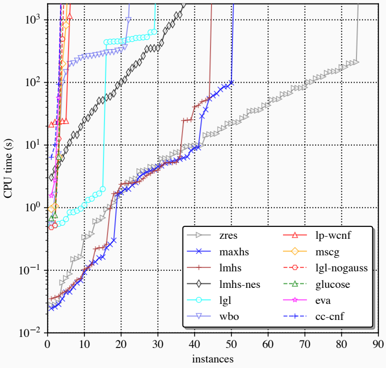


(a) cactus plot

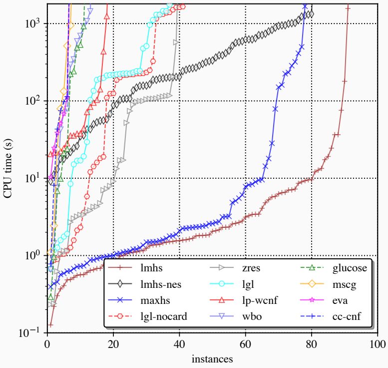


(b) *wbo* w/ and w/o $\mathcal{P}$ clauses

Performance on Urquhart and combined formulas



(a) URQ instances



(b) COMB instances

Overall performance

		glucose	lgl	lgl-no[3]	maxhs	lmhs	lmhs-nes	mscg	wbo	eva	lp-cnf	lp-wcnf	cc-cnf	cc-opb	zres
PHP-pw	(46)	7	29	7	***46***	***46***	29	***46***	10	***46***	***46***	***46***	6	5	10
PHP-sc	(46)	13	11	11	***46***	***46***	45	***46***	15	40	***46***	***46***	6	2	8
URQ	(84)	3	29	4	50	44	37	5	22	3	0	6	3	0	***84***
COMB	(96)	11	37	41	78	***91***	80	7	13	6	0	18	6	0	39
Total	(272)	34	106	63	220	***227***	191	104	60	95	92	116	21	7	141

[3]This represents *lgl-nogauss* for URQ and *lgl-nocard* for PHP-pw, PHP-sc, and COMB.

Summary and future work

Summary and future work

- solving SAT with DRE+MaxSAT

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

- more easy examples?

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

- more easy examples? any hard examples?

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

- more easy examples? any hard examples?
- similar transformations for other hard examples?

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

- more easy examples? any hard examples?
- similar transformations for other hard examples?
- integrating in a SAT solver?

Summary and future work

- solving SAT with DRE+MaxSAT
    - dual-rail (Horn) encoding
    - apply MaxSAT technology:
        - core-guided reasoning
        - MaxSAT resolution
    - refuting PHP in polynomial time
    - more examples (Urquhart and combined)

- more easy examples? any hard examples?
- similar transformations for other hard examples?
- integrating in a SAT solver?
- relation to known proof systems (TLR/GR/ER, CP, etc.)?

Summary and future work

- solving SAT with DRE+MaxSAT
  - dual-rail (Horn) encoding
  - apply MaxSAT technology:
    - core-guided reasoning
    - MaxSAT resolution
  - refuting PHP in polynomial time
  - more examples (Urquhart and combined)

- more easy examples? any hard examples?
- similar transformations for other hard examples?
- integrating in a SAT solver?
- relation to known proof systems (TLR/GR/ER, CP, etc.)?
- why is IHS so *good*?

Bonus: relation to known proof systems

Bonus: relation to known proof systems

- DRE+MaxSAT *p*-simulates TLR

- DRE+MaxSAT p-simulates TLR
- DRE+MaxSAT p-simulates GR

Bonus: relation to known proof systems

- DRE+MaxSAT p-simulates TLR
- DRE+MaxSAT p-simulates GR
- DRE+MaxSAT ~~p-simulates~~ CP

($*$ AAAI18 submission $*$)

Questions?