Hard Examples for Common Variable Decision Heuristics

Marc Vinyals

Technion
Haifa, Israel

Dagstuhl workshop on SAT and Interactions

DPLL

Algorithm 1: DPLL

```
while not solved do
    if conflict then backtrack()
    else if unit then propagate()
    else
        decide()
```

State: partial assignment

CDCL

Algorithm 2: CDCL

while *not solved* **do**
- **if** *conflict* **then** learn()
- **else if** *unit* **then** propagate()
- **else**
  - maybe forget()
  - maybe restart()
  - decide()

State: partial assignment & learned clauses

Resolution

$$\frac{C \vee v \qquad D \vee \bar{v}}{C \vee D}$$

$x \vee z$ $\quad$ $y \vee \bar{z}$ $\quad$ $x \vee \bar{y}$ $\quad$ $\bar{x} \vee \bar{y}$ $\quad$ $\bar{x} \vee y$

Resolution

$$\frac{C \lor v \qquad D \lor \bar{v}}{C \lor D}$$

$x \lor z$ $\qquad$ $y \lor \bar{z}$ $\qquad$ $x \lor \bar{y}$ $\qquad$ $\bar{x} \lor \bar{y}$ $\qquad$ $\bar{x} \lor y$

$x \lor y$

Resolution

$$\frac{C \vee v \qquad D \vee \overline{v}}{C \vee D}$$

$x \vee z$ $\quad$ $y \vee \overline{z}$ $\quad$ $x \vee \overline{y}$ $\quad$ $\overline{x} \vee \overline{y}$ $\quad$ $\overline{x} \vee y$

$x \vee y$

x

Resolution

$$\frac{C \lor v \qquad D \lor \bar{v}}{C \lor D}$$

$x \lor z$ $\quad$ $y \lor \bar{z}$ $\quad$ $x \lor \bar{y}$ $\quad$ $\bar{x} \lor \bar{y}$ $\quad$ $\bar{x} \lor y$

$x \lor y$

x

y

Resolution

$$\frac{C \vee v \qquad D \vee \overline{v}}{C \vee D}$$

$x \vee z$ $y \vee \overline{z}$ $x \vee \overline{y}$ $\overline{x} \vee \overline{y}$ $\overline{x} \vee y$

$x \vee y$

x

y

$\overline{x}$

Resolution

$$\frac{C \lor v \qquad D \lor \overline{v}}{C \lor D}$$

$x \lor z$ $\qquad$ $y \lor \overline{z}$ $\qquad$ $x \lor \overline{y}$ $\qquad$ $\overline{x} \lor \overline{y}$ $\qquad$ $\overline{x} \lor y$

$x \lor y$

x

y

$\overline{x}$

$\bot$

CDCL equivalent to Resolution

Observation [Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

CDCL equivalent to Resolution

Observation [Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem [Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

CDCL equivalent to Resolution

Observation [Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem [Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

Wait a minute...

Theorem [Atserias, Müller '19]

If a deterministic algorithm efficiently finds resolution proofs then $\mathrm{P} = \mathrm{NP}$

CDCL equivalent to Resolution

Observation [Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem [Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

Wait a minute…

Theorem [Atserias, Müller '19]

If a deterministic algorithm efficiently finds resolution proofs then $\mathrm{P} = \mathrm{NP}$

with non-deterministic variable decisions

CDCL equivalent to Resolution

Observation [Beame, Kautz, Sabharwal '04]

CDCL produces resolution proofs

Theorem [Pipatsrisawat, Darwiche '09]

CDCL can efficiently find resolution proofs

Wait a minute...

Theorem [Atserias, Müller '19]

If a deterministic algorithm efficiently finds resolution proofs then P = NP

with non-deterministic variable decisions

Also: CDCL with random decisions simulates bounded-width Resolution [Atserias, Fichte, Thurley '09].

Separation of CDCL vs Resolution

Theorem

There are formulas such that

- Resolution refutations of polynomial length
- Exponential time in CDCL with common variable decision heuristics

Separation of CDCL vs Resolution

Theorem

There are formulas such that

- Resolution refutations of polynomial length
- Exponential time in CDCL with common variable decision heuristics

Variable Decision Heuristics

Which literal do we pick next?

- Will lead to a conflict quickly.
- Was involved in conflicts recently.

Algorithm 2: CDCL

```
while not solved do
    if conflict then learn()
    else if unit then propagate()
    else
        maybe forget()
        maybe restart()
        decide()
```

Variable Decision Heuristics

Which literal do we pick next?

- Will lead to a conflict quickly.
- Was involved in conflicts recently.

VSIDS

- Give a score $q(x)$ to variable x.
- At each conflict
  - Bump $q' = q + 1$ if x involved.
  - Decay $q' = 0.95 \cdot q$ all variables.
- Pick variable with largest score

Algorithm 2: CDCL

```
while not solved do
    if conflict then learn()
    else if unit then propagate()
    else
        maybe forget()
        maybe restart()
        decide()
```

Variable Decision Heuristics

Which literal do we pick next?

- Will lead to a conflict quickly.
- Was involved in conflicts recently.

VSIDS

- Give a score $q(x)$ to variable x.
- At each conflict
  - Bump $q' = q + 1$ if x involved.
  - Decay $q' = 0.95 \cdot q$ all variables.
- Pick variable with largest score

Sign

- Last assigned.

Algorithm 2: CDCL

while *not solved* **do**
    if *conflict* **then** learn()
    else if *unit* **then** propagate()
    else
        maybe forget()
        maybe restart()
        decide()

Properties of VSIDS

- Each conflict
  - Bump $q' = q + 1$ if x involved.
  - Decay $q' = 0.95 \cdot q$ all variables.

Observation

A variable involved in a conflict is picked before a variable that never has.

Properties of VSIDS

- Each conflict
  - Bump $q' = q + 1$ if x involved.
  - Decay $q' = 0.95 \cdot q$ all variables.

Observation

A variable involved in a conflict is picked before a variable that never has.

Fine Print

Not true if finite precision.

Does hold if stable priority queue.

Separation of CDCL vs Resolution

Definition

A decision heuristic rewards conflicts if
a variable involved in a conflict is picked before a variable that never has.

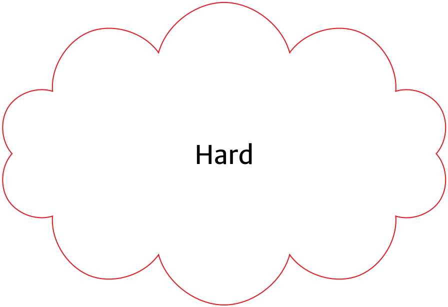
Theorem

There are formulas such that

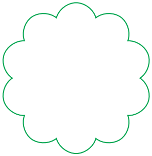
- Resolution refutations of polynomial length
- Exponential time in CDCL with conflict-rewarding heuristics

Intuition

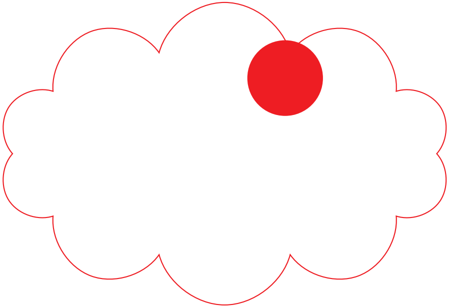
- Easy part + Hard part.



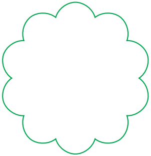
Intuition



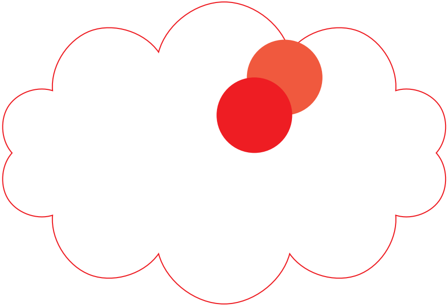
- Easy part + Hard part.
- Conflict in hard part $\implies$ More conflicts in hard part.



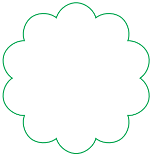
Intuition



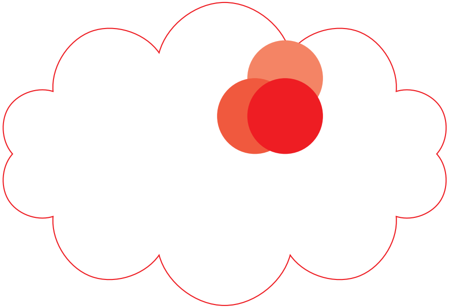
- Easy part + Hard part.
- Conflict in hard part $\Longrightarrow$
  More conflicts in hard part.



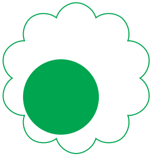
Intuition



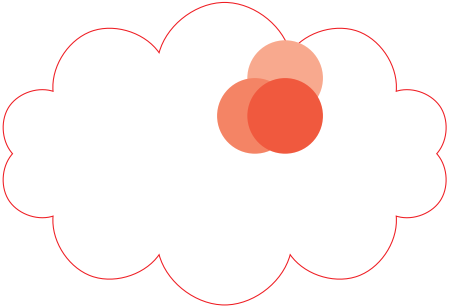
- Easy part + Hard part.
- Conflict in hard part $\implies$
  More conflicts in hard part.



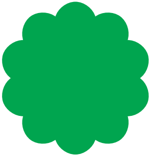
Intuition



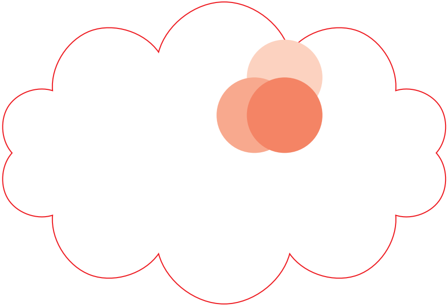
- Easy part + Hard part.
- Conflict in hard part $\Longrightarrow$ More conflicts in hard part.
- But hard formulas are global.
- Eventually stabilize.
- Then chance to hit easy formula.



Intuition

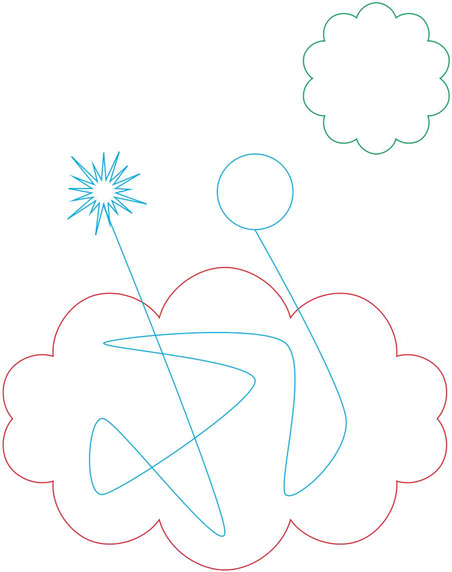


- Easy part + Hard part.
- Conflict in hard part $\Longrightarrow$ More conflicts in hard part.
- But hard formulas are global.
- Eventually stabilize.
- Then chance to hit easy formula.



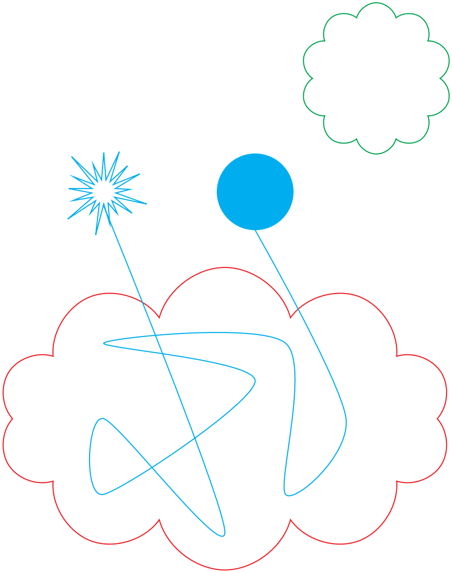
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



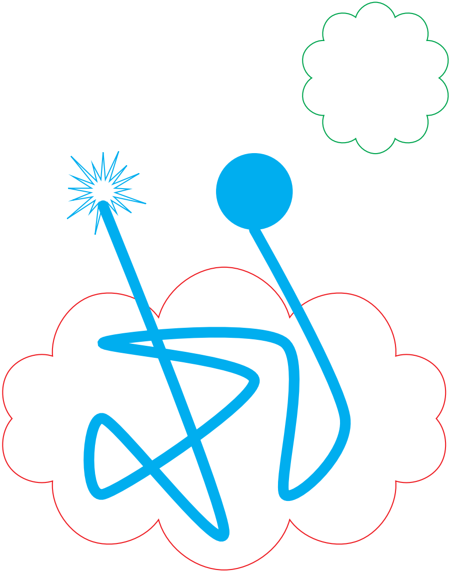
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



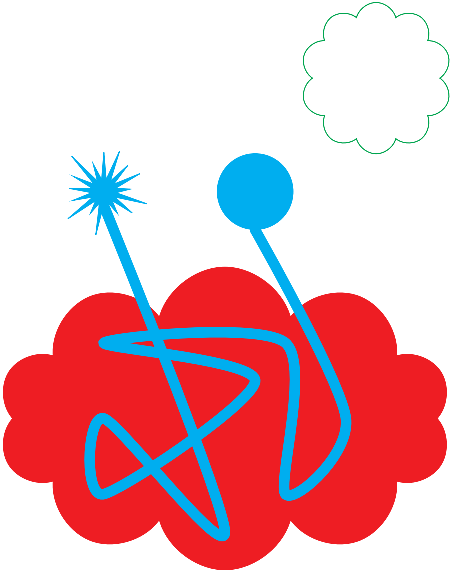
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



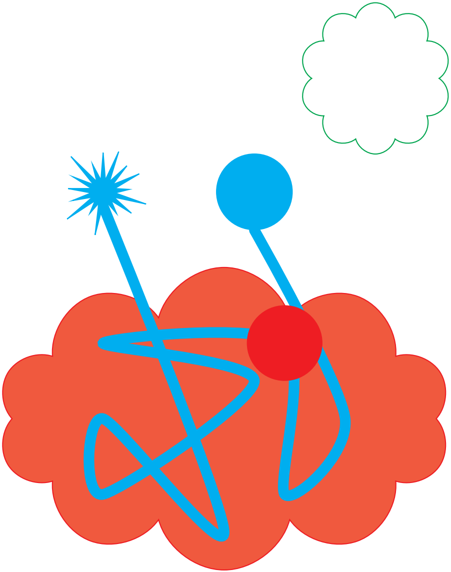
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



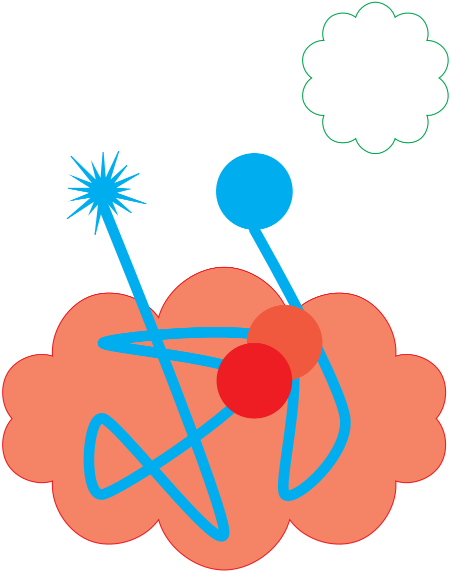
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.
- Solver stuck with hard variables!



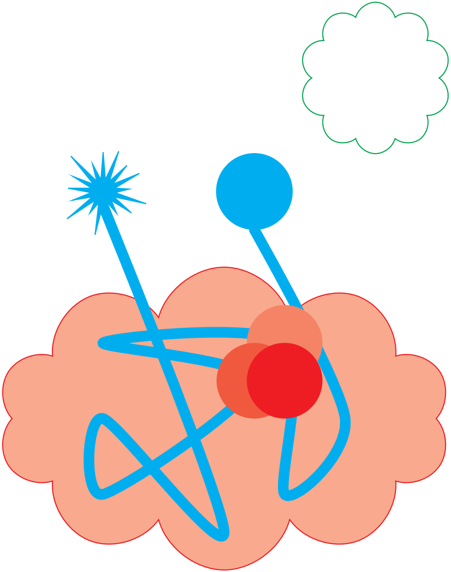
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.
- Solver stuck with hard variables!



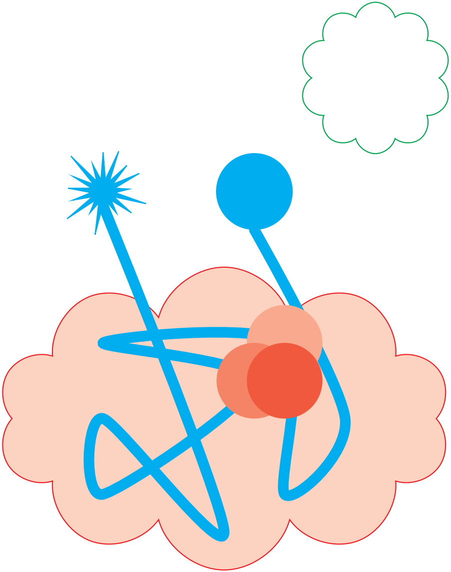
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.
- Solver stuck with hard variables!



Intuition (II)

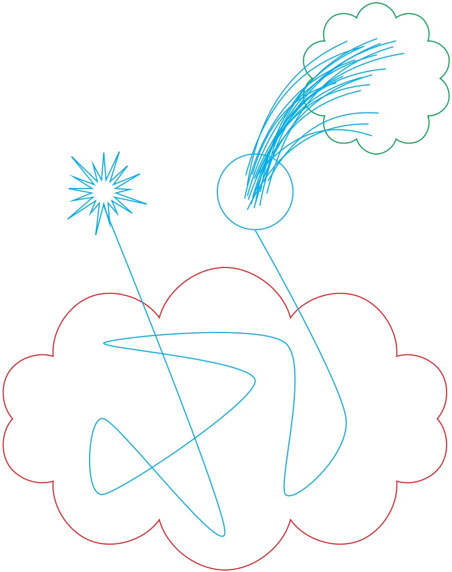
- Pitfall gadget produces a conflict involving all hard variables.
- Solver stuck with hard variables!



- But still $1/\mathrm{poly}$ probability of solving easy part first.

Intuition (III)

- Make easy variables lead to pitfall gadget.



Formula Description

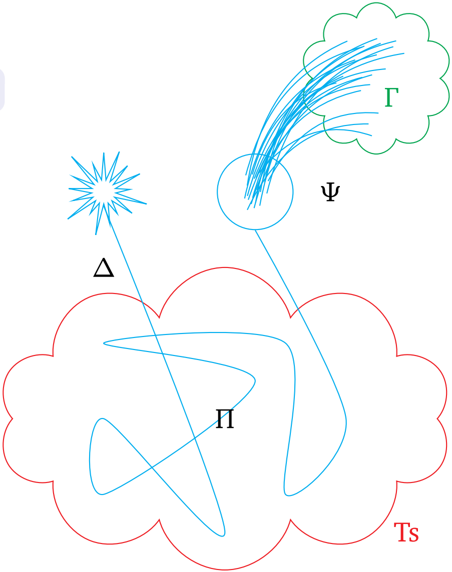
Pitfall Formula Φ

Variables

X	Hard
Y	Easy
Z	Auxiliary

Gadgets

$\mathrm{Ts}(X,Z)$	Padded Tseitin
$\Gamma(Y)$	Easy
$\Psi(Y,Z)$ & $\Pi(Z,X)$	Pitfall
$\Delta(Z)$	Tail



Proof Sketch

- Assume have a proof $\pi : \Phi \vdash \bot$ that does not use Γ clauses.
  In other words have a proof $\pi : (\Phi \setminus \Gamma) \vdash \bot$.

Proof Sketch

- Assume have a proof $\pi : \Phi \vdash \bot$ that does not use Γ clauses.
  In other words have a proof $\pi : (\Phi \setminus \Gamma) \vdash \bot$.

- Hit π with restriction ρ st $\rho(X) = *$ and ρ satisfies auxiliary gadgets.

Proof Sketch

- Assume have a proof $\pi : \Phi \vdash \bot$ that does not use Γ clauses.
  In other words have a proof $\pi : (\Phi \setminus \Gamma) \vdash \bot$.

- Hit π with restriction ρ st $\rho(X) = *$ and ρ satisfies auxiliary gadgets.

- Have a proof $\pi\restriction_\rho : (\Phi \setminus \Gamma)\restriction_\rho \vdash \bot$.
  In other words $\pi\restriction_\rho : \mathrm{Ts} \vdash \bot$.

Proof Sketch

- Assume have a proof $\pi : \Phi \vdash \perp$ that does not use Γ clauses.
  In other words have a proof $\pi : (\Phi \setminus \Gamma) \vdash \perp$.
- Hit π with restriction ρ st $\rho(X) = *$ and ρ satisfies auxiliary gadgets.
- Have a proof $\pi\restriction_\rho : (\Phi \setminus \Gamma)\restriction_\rho \vdash \perp$.
  In other words $\pi\restriction_\rho : \mathrm{Ts} \vdash \perp$.
- Hence π exponential.

Proof Sketch (II)

Need to ensure no conflicts use Γ clauses.
Define following solver states:

(a)

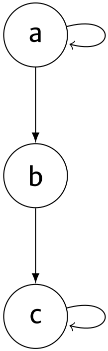
- No conflict
- No pair of Y variables assigned
- Enough Z variables unassigned

(b)

- (a) + a pair of Y variables assigned

(c)

- (a) + all X variables involved in a conflict

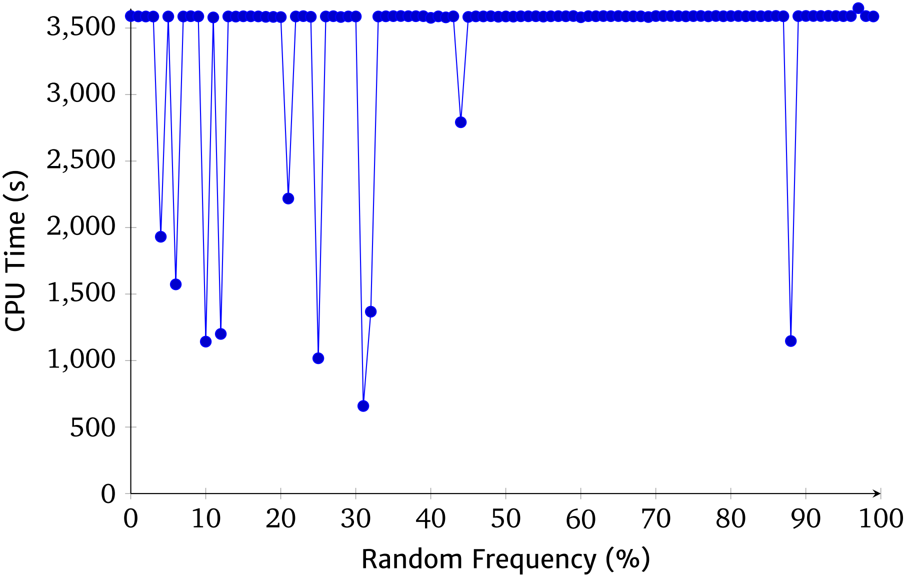


Experimental Results

Mean CPU time to solve (s)

Formula	CaDiCaL	Glucose	Maple CHB	Maple LRB	Static
Ts(45)	3331	754	621	424	3600
$\Phi(45, 6)$	2228	1917	600	2598	< 1
$\Phi(45, 8)$	1963	2273	607	2650	< 1
$\Phi(45, 10)$	2356	1818	689	2521	< 1
Ts(50)	3600	3600	3600	3600	3600
$\Phi(50, 6)$	3600	3600	3600	3600	< 1
$\Phi(50, 8)$	3600	3600	3600	3600	< 1
$\Phi(50, 10)$	3600	3600	3600	3600	< 1

Effects of Random Decisions



Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Open Problems

- CDCL with VSIDS vs CDCL with random decisions?
- Lower bound robust wrt score precision?
- Simpler construction?
- Abstract proof?

Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Open Problems

- CDCL with VSIDS vs CDCL with random decisions?
- Lower bound robust wrt score precision?
- Simpler construction?
- Abstract proof?

Thanks!