Hard Examples for Common Variable Decision Heuristics

Marc Vinyals

Technion
Haifa, Israel

Theoretical Foundations of SAT/SMT Solving

DPLL

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

```
Algorithm 1: DPLL
while not solved do
    if conflict then backtrack()
    else if unit then propagate()
    else branch()
```

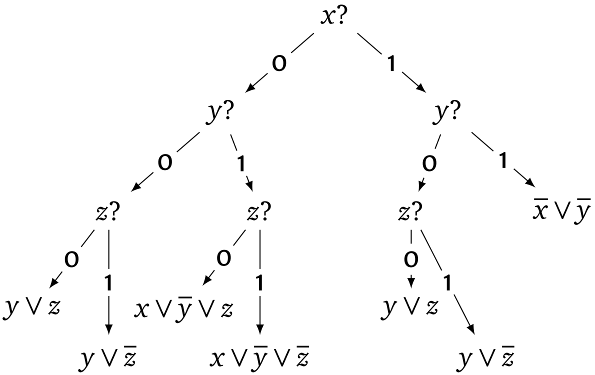
State: partial assignment

DPLL

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

```
Algorithm 1: DPLL
while not solved do
    if conflict then backtrack()
    else if unit then propagate()
    else branch()
```

State: partial assignment

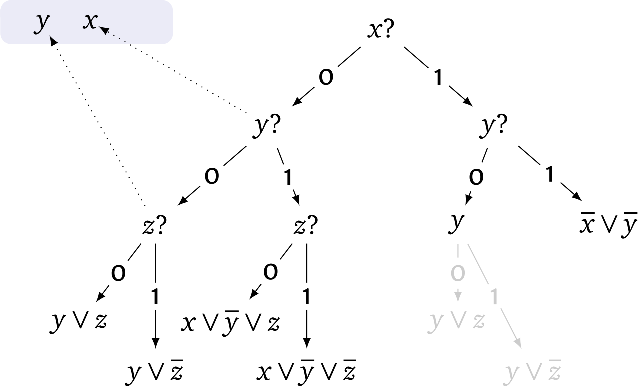

$x?$
0
1
$y?$
$y?$
0
1
0
1
$z?$
$z?$
$z?$
$\overline{x} \lor \overline{y}$
0
1
0
1
0
1
$y \lor z$
$y \lor \overline{z}$
$x \lor \overline{y} \lor z$
$x \lor \overline{y} \lor \overline{z}$
$y \lor z$
$y \lor \overline{z}$

CDCL

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

```
Algorithm 2: CDCL
while not solved do
    if conflict then learn()
    else if unit then propagate()
    else branch()
```

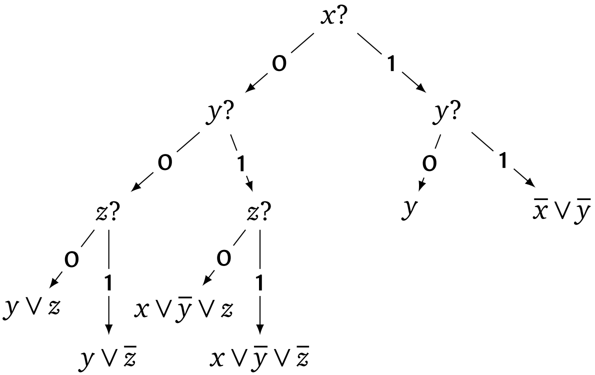
State: partial assignment
& learned clauses



Resolution

- Interpret CDCL run as resolution proof

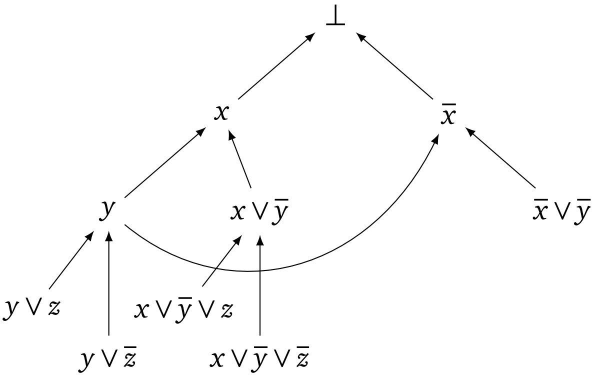
$$\frac{C \lor v \qquad D \lor \overline{v}}{C \lor D}$$


$x?$
0
1
$y?$
$y?$
0
1
0
1
$z?$
$z?$
y
$\overline{x} \lor \overline{y}$
0
1
0
1
$y \lor z$
$y \lor \overline{z}$
$x \lor \overline{y} \lor z$
$x \lor \overline{y} \lor \overline{z}$

Resolution

- Interpret CDCL run as resolution proof

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



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

CDCL vs Resolution

- CDCL implicit proofs are in resolution form
- DPLL proofs only in weaker “tree-like” resolution form
- Is CDCL as powerful as general resolution?

CDCL vs Resolution

- CDCL implicit proofs are in resolution form
- DPLL proofs only in weaker “tree-like” resolution form
- Is CDCL as powerful as general resolution?
- Partial results in 2000s [Beame, Kautz, Sabharwal ’04] [Van Gelder ’05] [Hertel, Bacchus, Pitassi, Van Gelder ’08] [Buss, Hoffmann, Johannsen ’08]
- Yes (under natural model) [Pipatsrisawat, Darwiche ’09]
- Assumptions: [Atserias, Fichte, Thurley ’09]
  - Optimal variable choices
  - Clauses not thrown away
  - Frequent restarts
  - Standard learning

CDCL equivalent to Resolution

Theorem [Pipatsrisawat, Darwiche ’09]

CDCL can efficiently find resolution proofs

CDCL equivalent to Resolution

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

Theorem [Pipatsrisawat, Darwiche ’09]

CDCL can efficiently find resolution proofs

with non-deterministic variable decisions

Wait a minute. . .

Theorem [Atserias, Müller ’19]

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

CDCL equivalent to Resolution

Theorem [Pipatsrisawat, Darwiche ’09]

CDCL can efficiently find resolution proofs

with non-deterministic variable decisions

Also: CDCL with random decisions simulates bounded-width Resolution

[Atserias, Fichte, Thurley ’09]

Wait a minute…

Theorem [Atserias, Müller ’19]

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

Separation of CDCL vs Resolution

Theorem [V '20]

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 [V '20]

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 branch()
```

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 branch()
```

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 branch()
```

Properties of VSIDS

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

Observation

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

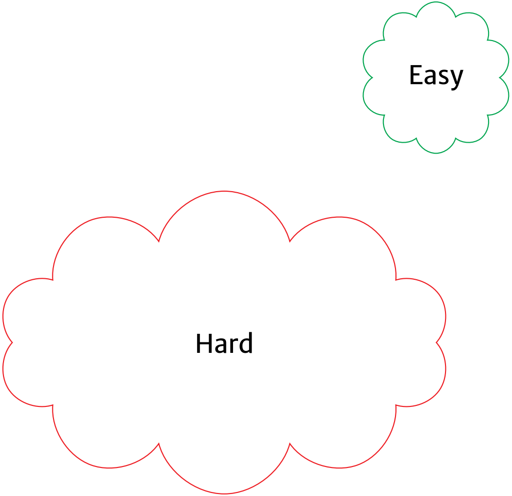
Fine Print

Not true if finite precision.

Does hold if stable priority queue.

Intuition

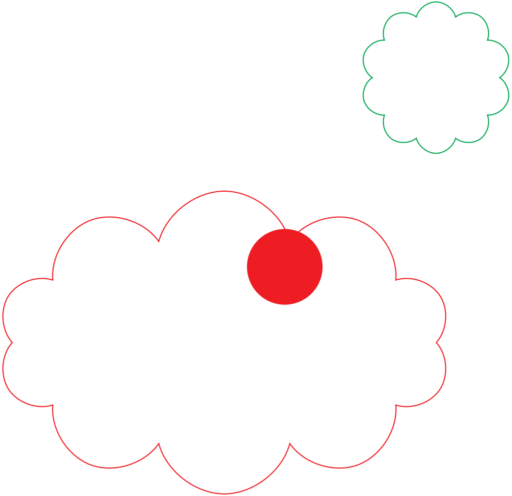
- Easy part + Hard part.


Easy

Hard

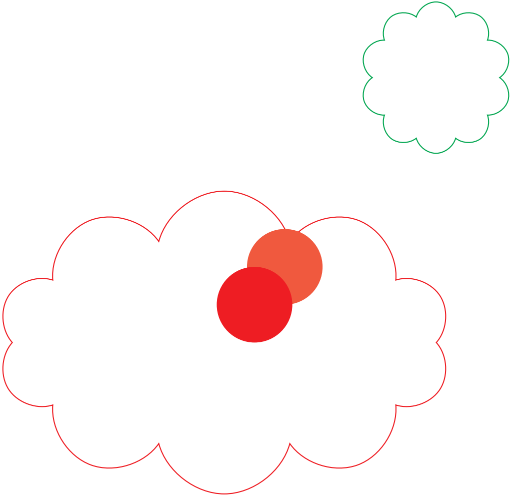
Intuition

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



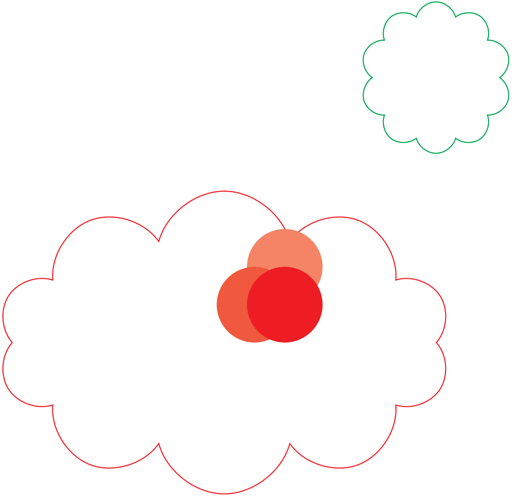
Intuition

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



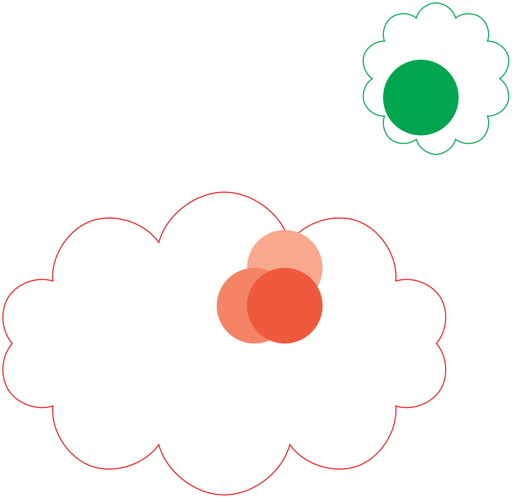
Intuition

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



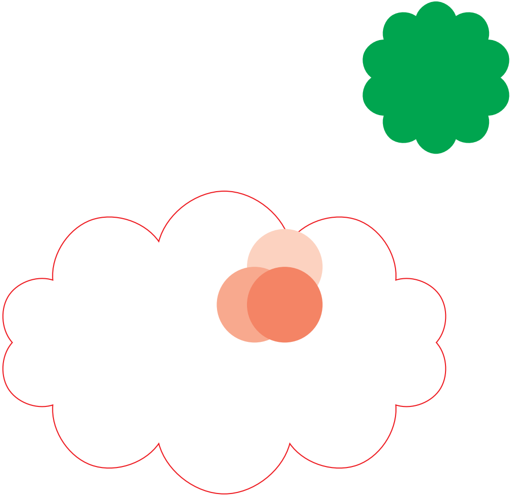
Intuition

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



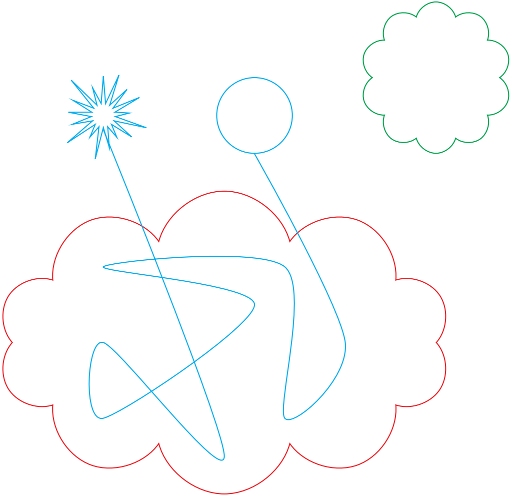
Intuition

- 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.



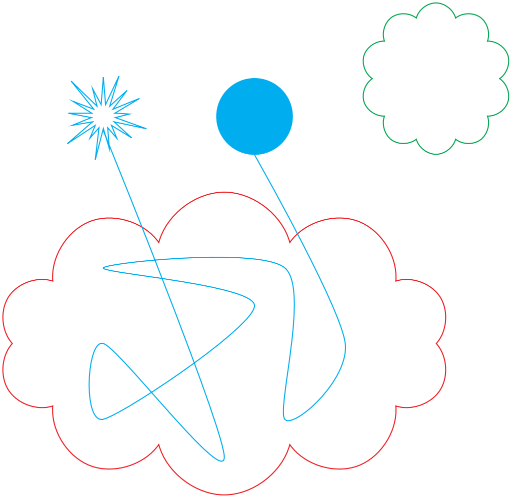
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



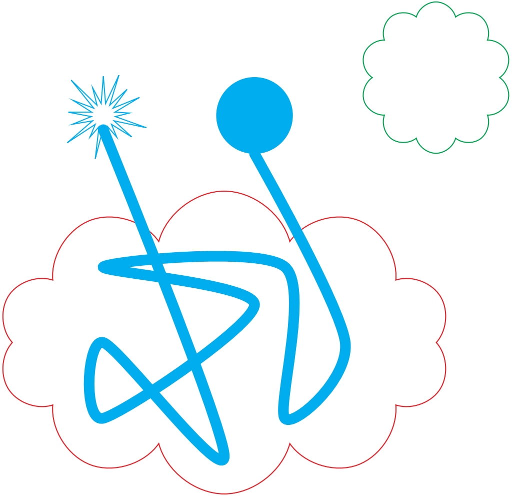
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



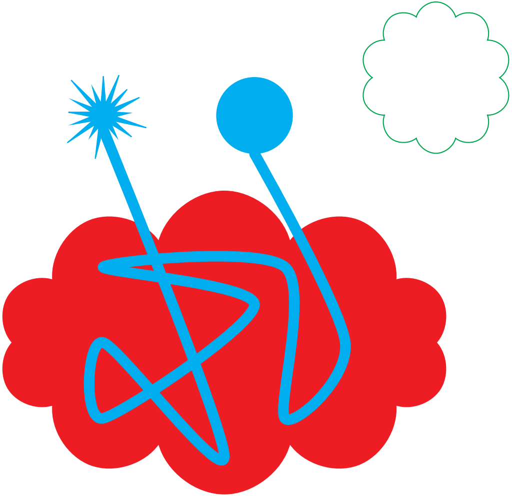
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



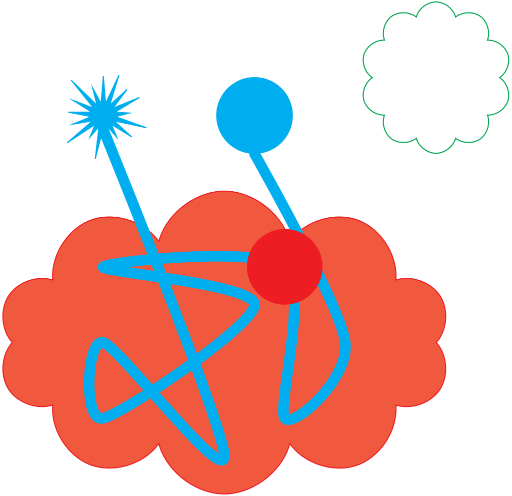
Intuition (II)

- Pitfall gadget produces a conflict involving all hard variables.



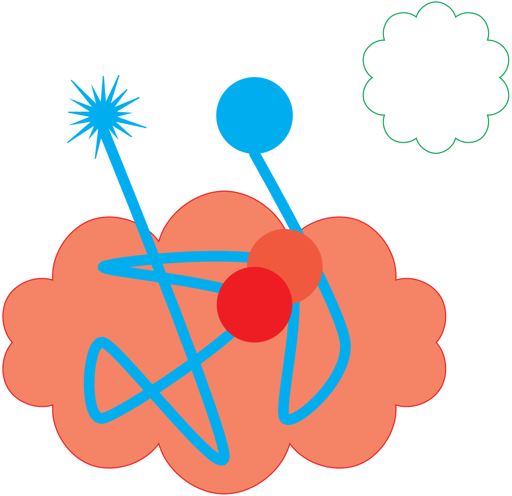
Intuition (II)

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



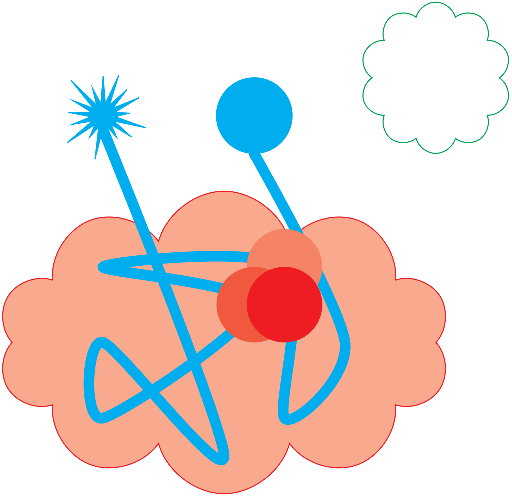
Intuition (II)

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



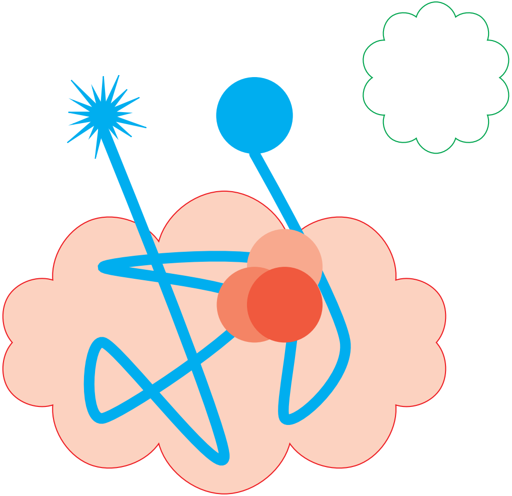
Intuition (II)

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



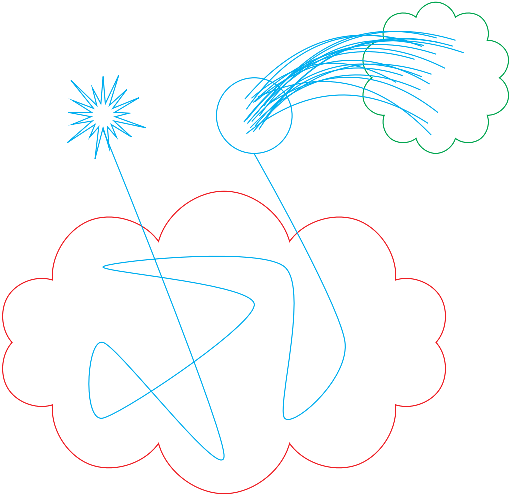
Intuition (II)

- 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.



Separation of CDCL vs Resolution

Property

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

- Result holds for any conflict-rewarding heuristics: VMTF, VSIDS*, CHB, LRB*

Separation of CDCL vs Resolution

Property

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

- Result holds for any conflict-rewarding heuristics: VMTF, VSIDS*, CHB, LRB*
- Result holds with and without restarts
- But not if restarts clear score (see Ian's talk)

Experiments

Mean CPU time to solve (s)

	CaDiCaL	Glucose	MapleSAT	MapleSAT	
Formula	VMTF	VSIDS	CHB	LRB	Static
Hard(45)	3331	754	621	424	3600
Hard(50)	3600	3600	3600	3600	3600

Experiments

Mean CPU time to solve (s)

	CaDiCaL	Glucose	MapleSAT	MapleSAT	
Formula	VMTF	VSIDS	CHB	LRB	Static
Hard(45)	3331	754	621	424	3600
Pitfall(45)	1963	2273	607	2650	< 1
Hard(50)	3600	3600	3600	3600	3600
Pitfall(50)	3600	3600	3600	3600	< 1

Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Open Problems

- Proof robust wrt score precision?
- Simpler construction?
- Improve VSIDS?

Take Home

Result

- CDCL with VSIDS not equivalent to Resolution

Open Problems

- Proof robust wrt score precision?
- Simpler construction?
- Improve VSIDS?

Thanks!