The Impact of Bounded Variable Elimination on Solving Pigeonhole Formulas

Joseph E. Reeves and Marijn J. H. Heule

Carnegie Mellon University

Introduction

Bounded variable elimination (BVE) presented by Eén and Biere in 2005 is used by every state-of-the-art CDCL solver

An experimental study revealed slowdowns with BVE enabled for the 2020 SAT Competition winner KISSAT

We examined the impact of different variable elimination orderings on solving pigeon hole formulas

We found that different variable scoring strategies caused some solvers to be more stable, but some elimination orderings were generally hard

Bounded Variable Elimination

Definition (Resolution)

Given two clauses $C_1 = x \vee a_1 \vee ... \vee a_n$ and $C_2 = \overline{x} \vee b_1 \vee ... \vee b_m$, resolution ($C_1 \otimes C_2$) returns the *resolvent* $a_1 \vee ... \vee a_n \vee b_1 \vee ... \vee b_m$

Definition (Variable Elimination by Distribution) [DavisPutnam'60]

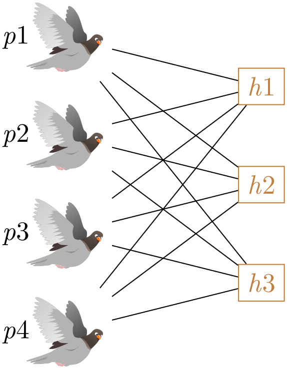
Replace all clauses containing x (S_x) and clauses containing $\overline{x}$ ($S_{\bar{x}}$) by the set:

$$\{C_1 \otimes C_2 | C_1 \in S_x, C_2 \in S_{\bar{x}}\}$$

Definition (Bounded Variable Elimination) [EénBiere'05]

Only eliminate when fewer clauses are added than deleted (or some other bound), made possible through elimination *by substitution* and gate extraction

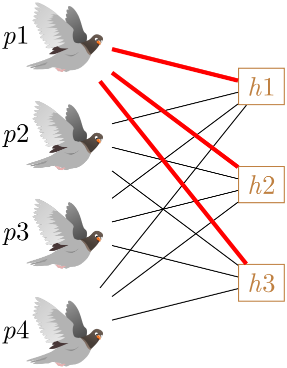
Pigeonhole Problem



Definition

- Place $n + 1$ pigeons into n holes
- Fully connected $K_{n,n+1}$
- Resolution proofs exponential

Pigeonhole Problem



ALO $(p1)$

$p_{1,1} \vee p_{1,2} \vee p_{1,3}$

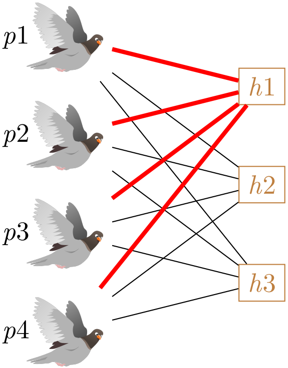
Definition

- Place $n + 1$ pigeons into n holes
- Fully connected $K_{n,n+1}$
- Resolution proofs exponential

Sparse

- At Least One (ALO) for pigeons

Pigeonhole Problem



AMO ($h1$)
$Pairwise(p_{1,1}, p_{2,1}, p_{3,1}, p_{4,1})$

Definition

- Place $n + 1$ pigeons into n holes
- Fully connected $K_{n,n+1}$
- Resolution proofs exponential

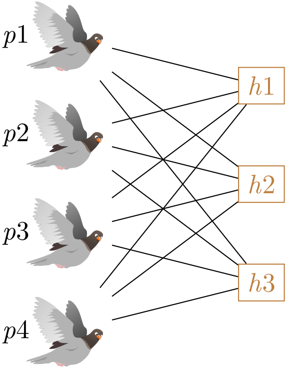
Sparse

- At Least One (ALO) for pigeons
- At Most One (AMO) for holes

Full

- ALO, AMO for both pigeons and holes

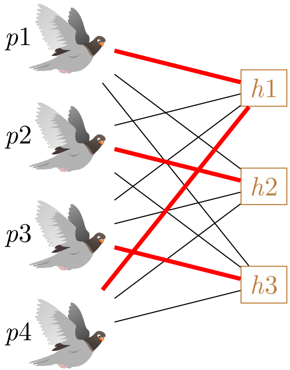
Pigeonhole Problem



Elimination

- Eliminate $n+1$ variables from independent pigeons
- Resolve the n binary clauses with the ALO disjunction to produce n new clauses
- Select variables (h): $p_{i,((i-1)\%h)+1}$ for $1 \leq i \leq n+1$
- Can only eliminate n variables from independent pigeons and independent holes for **Full** encoding

Pigeonhole Problem



$h = 3 \ (p_{1,1}, p_{2,2}, p_{3,3}, p_{4,1})$

Elimination

- Eliminate $n + 1$ variables from independent pigeons
- Resolve the n binary clauses with the ALO disjunction to produce n new clauses
- Select variables (h): $p_{i,((i-1)\%h)+1}$ for $1 \leq i \leq n + 1$
- Can only eliminate n variables from independent pigeons and independent holes for **Full** encoding

Top Tier Solvers and Competition Winners*

Kissat/CaDiCaL Focus on in-processing (BVE happens throughout)

Kissat20* Unsat/sat mode with EVSIDS/VMTF switching

Kissat21 Bumping during on-the-fly-strengthening, and UIP shrinking

CaDiCaL* C++ version of KISSAT (2020 version)

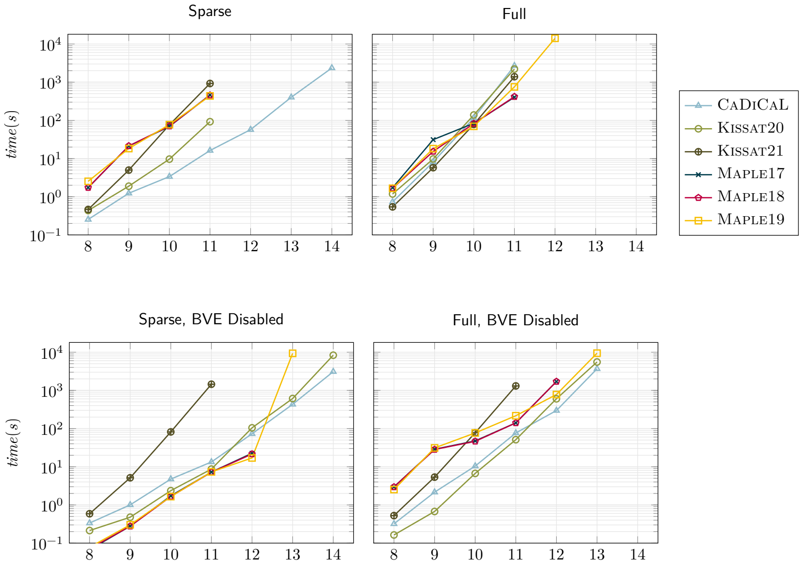
Maples Distance heuristic first 50,000 conflicts then LRB/VSIDS

Maple17* Learnt clause minimization

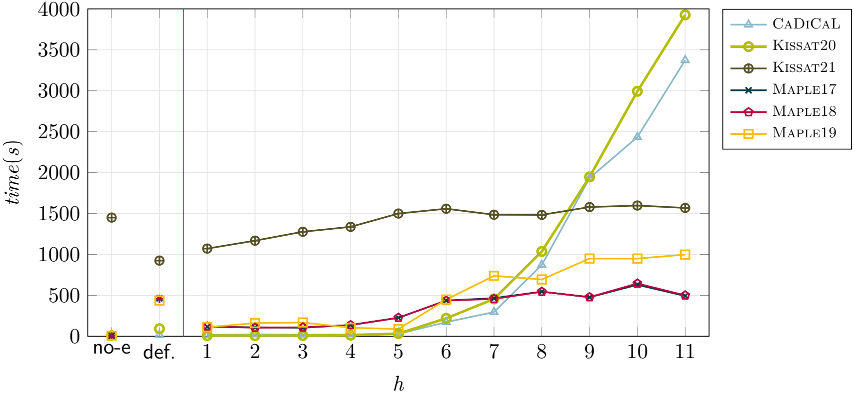
Maple18* Chronological backtracking

Maple19* Duplicate learnt clause tier strategy, and modifies VSIDS/LRB switching heuristic

Pigeonhole formulas (log plots)

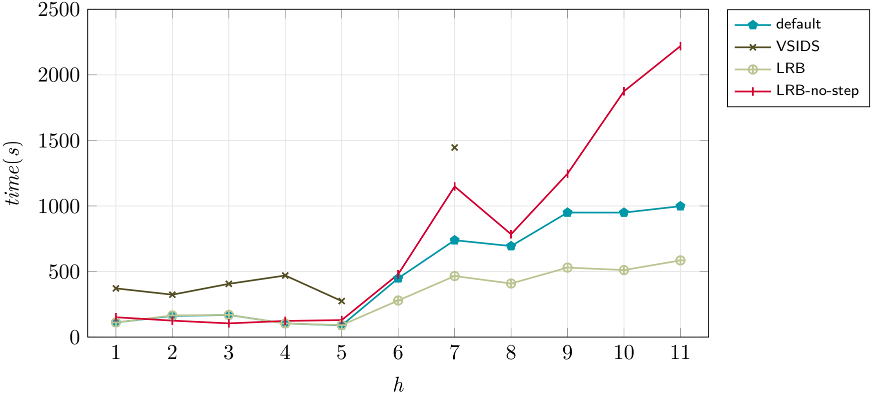


Pigeonhole BVE Instances



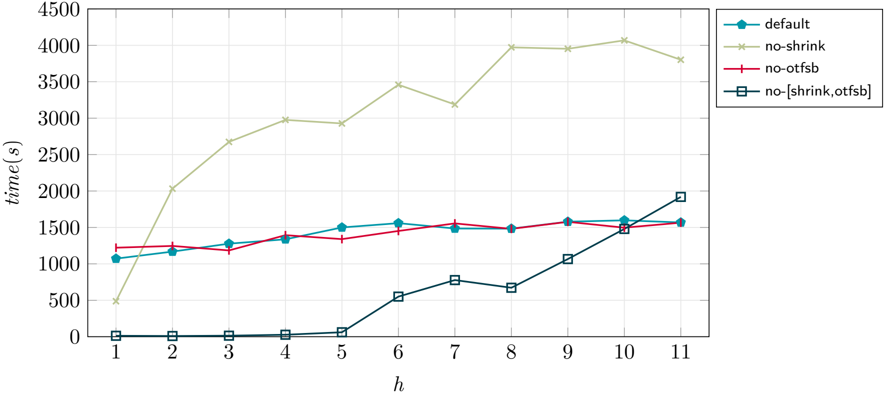
- 18000 second timeout over BVE instances for $n = 11$
- some solvers are stable, others have significant performance loss
- formulas get harder in general for $h > 5$
- $h = 11$ is elimination ordering forced in **Full** encoding

Maple19 BVE Instances for $n = 11$



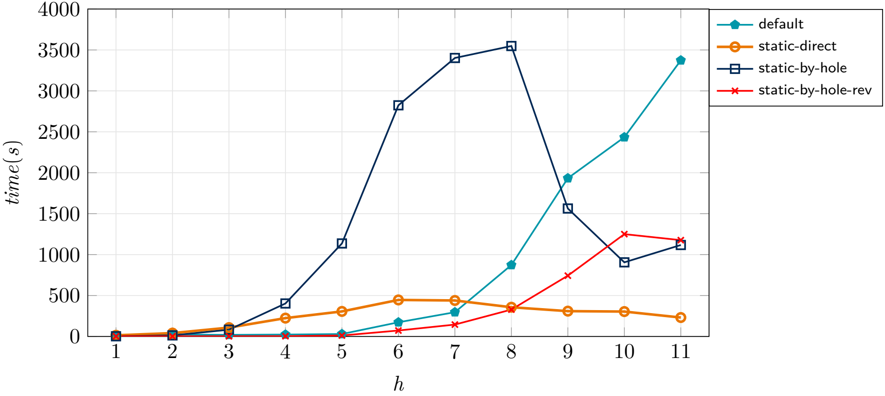
- VSIDS times out on many formulas $h > 6$
- LRB-step reduces importance of newly learned information over time (ML technique)
- LRB-step important to stabilize decision ordering

KISSAT21 BVE Instances for $n = 11$



- How to return to KISSAT20 performance?
- Answer: disabling UIP-shrink and bumping during on-the-fly-strengthening (otfsb)
- Disabling options individually not helpful

CaDiCaL BVE Instances for $n = 11$



- static-direct shows stable good performance (best for $h > 9$)
- Reversing static-by-hole shows best performance for $h < 9$
- Need a good static ordering to beat default configuration

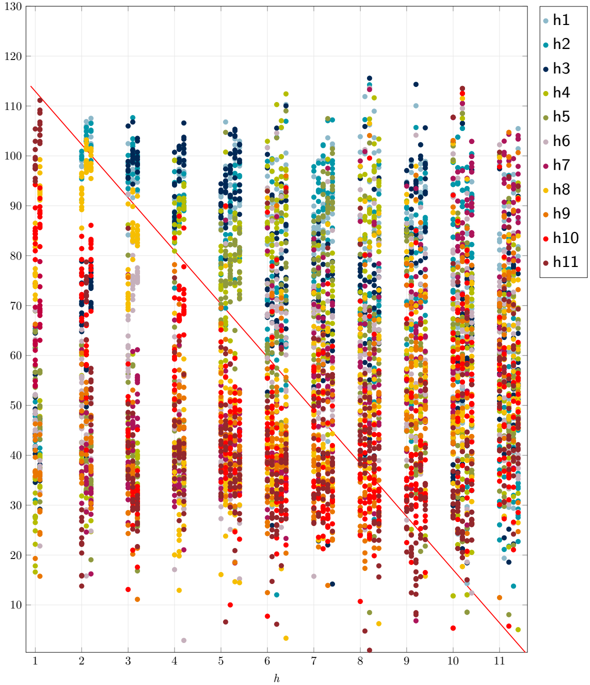
Conclusion

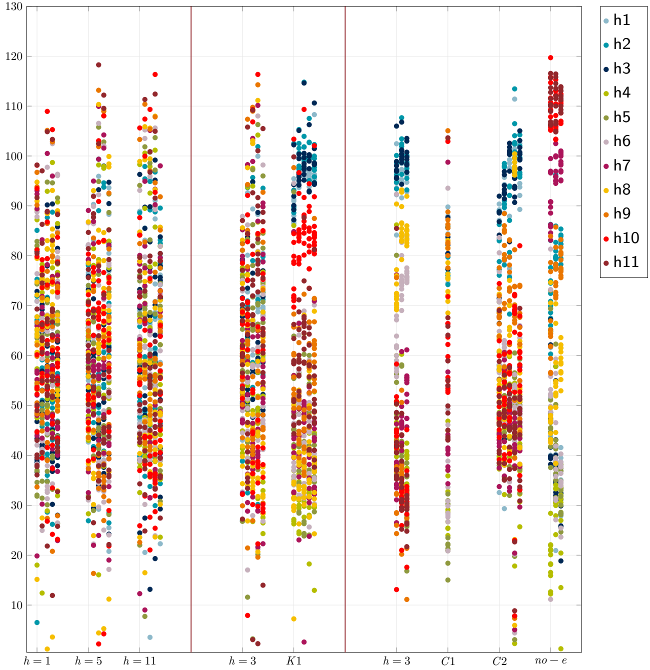
BVE can have a large negative impact on performance under certain variable orderings

Scoring strategies can mitigate this affect by producing more static decision orders

Avoiding elimination for exclusively binary clauses may also be helpful

But, solvers are complex and it is hard to generalize from specific formulas


130
120
110
100
90
80
70
60
50
40
30
20
10
1
2
3
4
5
6
7
8
9
10
11
h
h1
h2
h3
h4
h5
h6
h7
h8
h9
h10
h11


130
120
110
100
90
80
70
60
50
40
30
20
10
h1
h2
h3
h4
h5
h6
h7
h8
h9
h10
h11
h = 1
h = 5
h = 11
h = 3
K1
h = 3
C1
C2
no − e