Effective Auxiliary Variables via Structured Reencoding

Andrew Haberlandt*, Harrison Green*, Marijn Heule

Carnegie Mellon University

*equal contribution

Potential of auxiliary variables…

Potential of auxiliary variables…

…in encodings:

Table 1. Comparison of different encodings for $\leq k\,(x_1, \ldots, x_n)$.

Encoding	#clauses	#aux. vars	decided
Naïve	$\binom{n}{k+1}$	0	immediately
Sequential unary counter ($\mathrm{LT}_{\mathrm{SEQ}}^{n,k}$)	$\mathcal{O}(n \cdot k)$	$\mathcal{O}(n \cdot k)$	by unit prop.
Parallel binary counter ($\mathrm{LT}_{\mathrm{PAR}}^{n,k}$)	$7n - 3\lfloor \log n \rfloor - 6$	$2n - 2$	by search
Bailleux & Boufkhad [3]	$\mathcal{O}(n^2)$	$\mathcal{O}(n \cdot \log n)$	by unit prop.
Warners [4]	$8n$	$2n$	by search

Sinz, Carsten. "Towards an optimal CNF encoding of boolean cardinality constraints." International conference on principles and practice of constraint programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

Potential of auxiliary variables…

…in encodings:

Table 1. Comparison of different encodings for $\leq k\,(x_1, \ldots, x_n)$.

Encoding	#clauses	#aux. vars	decided
Naïve	$\binom{n}{k+1}$	0	immediately
Sequential unary counter ($\mathrm{LT}^{n,k}_{\mathrm{SEQ}}$)	$\mathcal{O}(n \cdot k)$	$\mathcal{O}(n \cdot k)$	by unit prop.
Parallel binary counter ($\mathrm{LT}^{n,k}_{\mathrm{PAR}}$)	$7n - 3\lfloor \log n \rfloor - 6$	$2n - 2$	by search
Bailleux & Boufkhad [3]	$\mathcal{O}(n^2)$	$\mathcal{O}(n \cdot \log n)$	by unit prop.
Warners [4]	$8n$	$2n$	by search

Sinz, Carsten. "Towards an optimal CNF encoding of boolean cardinality constraints." International conference on principles and practice of constraint programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

…for proofs:

SIGACT News 28 Oct.-Dec. 1976

A SHORT PROOF OF THE PIGEON HOLE PRINCIPLE
USING EXTENDED RESOLUTION

Stephen A. Cook

Potential of auxiliary variables…

…in encodings:

Table 1. Comparison of different encodings for $\leq k\ (x_1, \ldots, x_n)$.

Encoding	#clauses	#aux. vars	decided
Naïve	$\binom{n}{k+1}$	0	immediately
Sequential unary counter ($\mathrm{LT}^{n,k}_{\mathrm{SEQ}}$)	$\mathcal{O}(n \cdot k)$	$\mathcal{O}(n \cdot k)$	by unit prop.
Parallel binary counter ($\mathrm{LT}^{n,k}_{\mathrm{PAR}}$)	$7n - 3\lfloor \log n \rfloor - 6$	$2n - 2$	by search
Bailleux & Boufkhad [3]	$\mathcal{O}(n^2)$	$\mathcal{O}(n \cdot \log n)$	by unit prop.
Warners [4]	$8n$	$2n$	by search

Sinz, Carsten. "Towards an optimal CNF encoding of boolean cardinality constraints." International conference on principles and practice of constraint programming. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

…for proofs:

SIGACT News 28 Oct.-Dec. 1976

A SHORT PROOF OF THE PIGEON HOLE PRINCIPLE
USING EXTENDED RESOLUTION

Stephen A. Cook

…automatically …

Bounded Variable Addition (Manthey et al., 2012) reduces formulas by adding auxiliary variables

$$a \lor p$$

$$a \lor q$$

$$a \lor r$$

$$b \lor p$$

$$b \lor q$$

$$b \lor r$$

Manthey, Norbert, Marijn JH Heule, and Armin Biere. "Automated reencoding of boolean formulas." *Haifa Verification Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Bounded Variable Addition (Manthey et al., 2012) reduces formulas by adding auxiliary variables

$$a \lor p$$
$$a \lor q$$
$$a \lor r$$
$$b \lor p$$
$$b \lor q$$
$$b \lor r$$

$$\longrightarrow$$

$$\boldsymbol{x} \lor p$$
$$\boldsymbol{x} \lor q$$
$$\boldsymbol{x} \lor r$$
$$\overline{\boldsymbol{x}} \lor a$$
$$\overline{\boldsymbol{x}} \lor b$$

Manthey, Norbert, Marijn JH Heule, and Armin Biere. "Automated reencoding of boolean formulas." *Haifa Verification Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Bounded Variable Addition (Manthey et al., 2012) reduces formulas by adding auxiliary variables

$$a \lor p$$
$$a \lor q$$
$$a \lor r$$
$$b \lor p$$
$$b \lor q$$
$$b \lor r$$

$\longrightarrow$

$$\boldsymbol{x} \lor p$$
$$\boldsymbol{x} \lor q$$
$$\boldsymbol{x} \lor r$$
$$\overline{\boldsymbol{x}} \lor a$$
$$\overline{\boldsymbol{x}} \lor b$$

	$\boldsymbol{x} \lor p$	$\boldsymbol{x} \lor q$	$\boldsymbol{x} \lor r$
$\overline{\boldsymbol{x}} \lor a$	$a \lor p$	$a \lor q$	$a \lor r$
$\overline{\boldsymbol{x}} \lor b$	$b \lor p$	$b \lor q$	$b \lor r$

Manthey, Norbert, Marijn JH Heule, and Armin Biere. "Automated reencoding of boolean formulas." *Haifa Verification Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

Bounded Variable Addition (Manthey et al., 2012) reduces formulas by adding auxiliary variables

$a \lor p$

$a \lor q$

$a \lor r$

$b \lor p$

$b \lor q$

$b \lor r$

$\longrightarrow$

$x \lor p$

$x \lor q$

$x \lor r$

$\overline{x} \lor a$

$\overline{x} \lor b$

	$x \lor p$	$x \lor q$	$x \lor r$
$\overline{x} \lor a$	$\boxed{a \lor p}$	$a \lor q$	$a \lor r$
$\overline{x} \lor b$	$b \lor p$	$b \lor q$	$b \lor r$

$$(\overline{x} \lor a) \otimes (x \lor p) = a \lor p$$

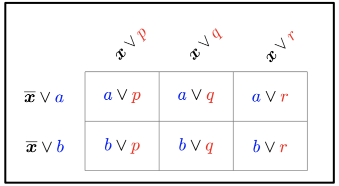
Manthey, Norbert, Marijn JH Heule, and Armin Biere. "Automated reencoding of boolean formulas." *Haifa Verification Conference*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

1

	$\boldsymbol{x} \vee p$	$\boldsymbol{x} \vee q$	$\boldsymbol{x} \vee r$
$\overline{\boldsymbol{x}} \vee a$	$a \vee p$	$a \vee q$	$a \vee r$
$\overline{\boldsymbol{x}} \vee b$	$b \vee p$	$b \vee q$	$b \vee r$

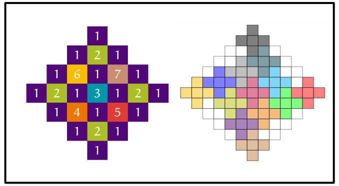
Bounded Variable Addition greedily constructs auxiliary variables

1



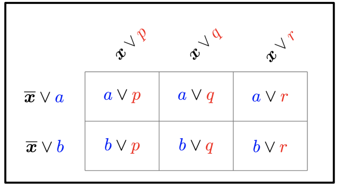
Bounded Variable Addition greedily constructs auxiliary variables

2



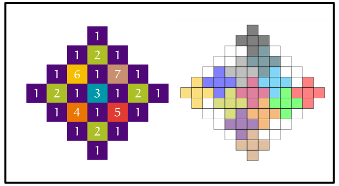
Packing k-coloring Problem - **auxiliary variables represent regions**

1



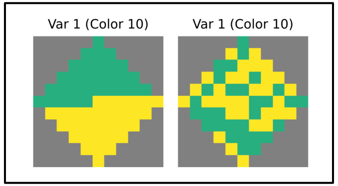
Bounded Variable Addition greedily constructs auxiliary variables

2



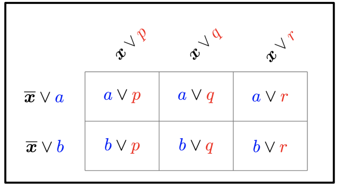
Packing k-coloring Problem - **auxiliary variables represent regions**

3



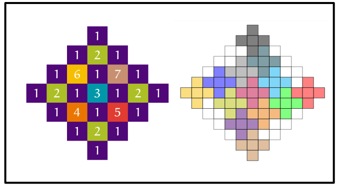
Randomization causes BVA to generate ineffective auxiliary variables

1



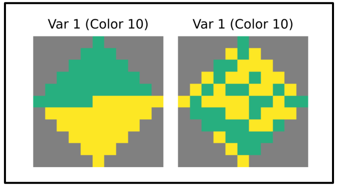
Bounded Variable Addition greedily constructs auxiliary variables

2



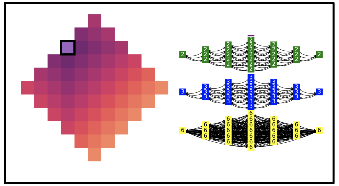
Packing k-coloring Problem - **auxiliary variables represent regions**

3

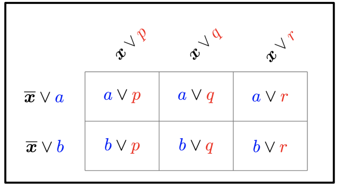


Randomization causes BVA to generate ineffective auxiliary variables

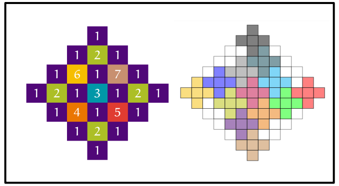
4



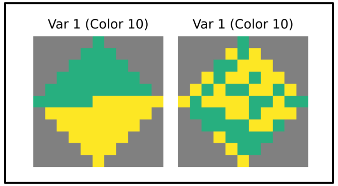
SBVA improves BVA with a **structure-based tiebreaking heuristic**

1 

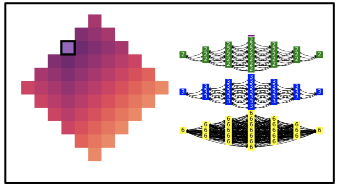
Bounded Variable Addition greedily constructs auxiliary variables

2 

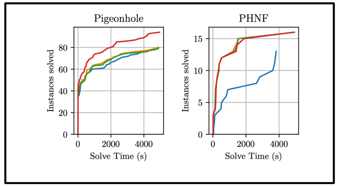
Packing k-coloring Problem - **auxiliary variables represent regions**

3 

Randomization causes BVA to generate ineffective auxiliary variables

4 

SBVA improves BVA with a **structure-based tiebreaking heuristic**

5 

SBVA is effective on a **diverse competition-style benchmark**

BVA finds grids of clauses in an iterative, greedy manner

	$\boldsymbol{x} \lor p$	$\boldsymbol{x} \lor q$	$\boldsymbol{x} \lor r$
$\overline{\boldsymbol{x}} \lor a$	$a \lor p$	$a \lor q$	$a \lor r$
$\overline{\boldsymbol{x}} \lor b$	$b \lor p$	$b \lor q$	$b \lor r$

BVA finds grids of clauses in an iterative, greedy manner

	p	q	r	$s \lor t$
a	$a \lor p$	$a \lor q$	$a \lor r$	$a \lor s \lor t$
b	$b \lor p$	$b \lor q$	$b \lor r$	
c		$c \lor q$		

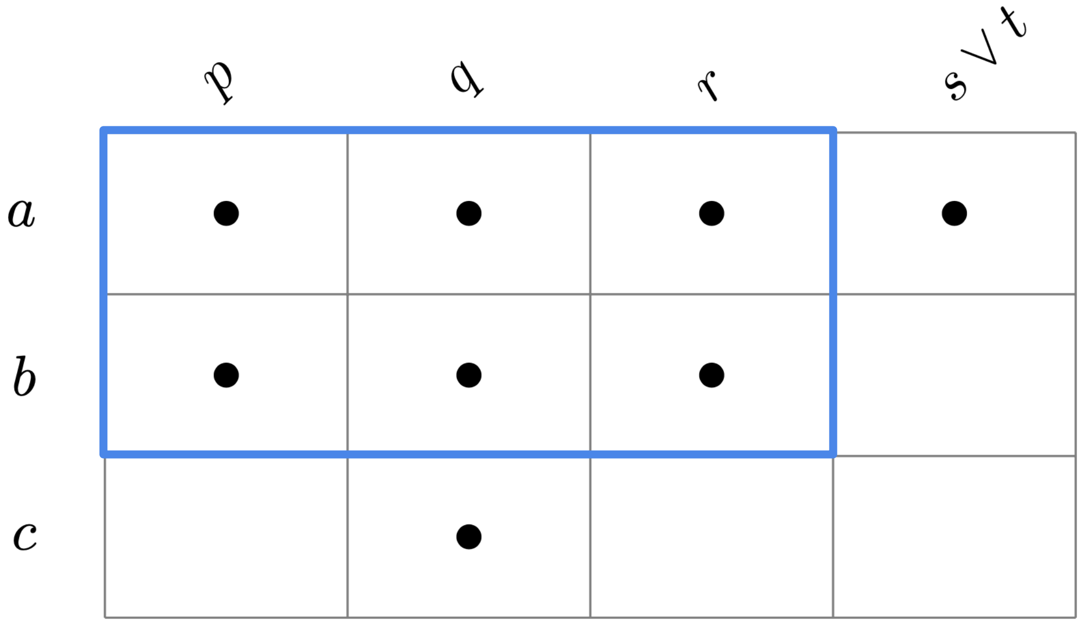
BVA finds grids of clauses in an iterative, greedy manner

	p	q	r	$s \vee t$
a	●	●	●	●
b	●	●	●	
c		●		

BVA finds grids of clauses in an iterative, greedy manner

	p	q	r	$s \vee t$
a	●	●	●	●
b	●	●	●	
c		●		

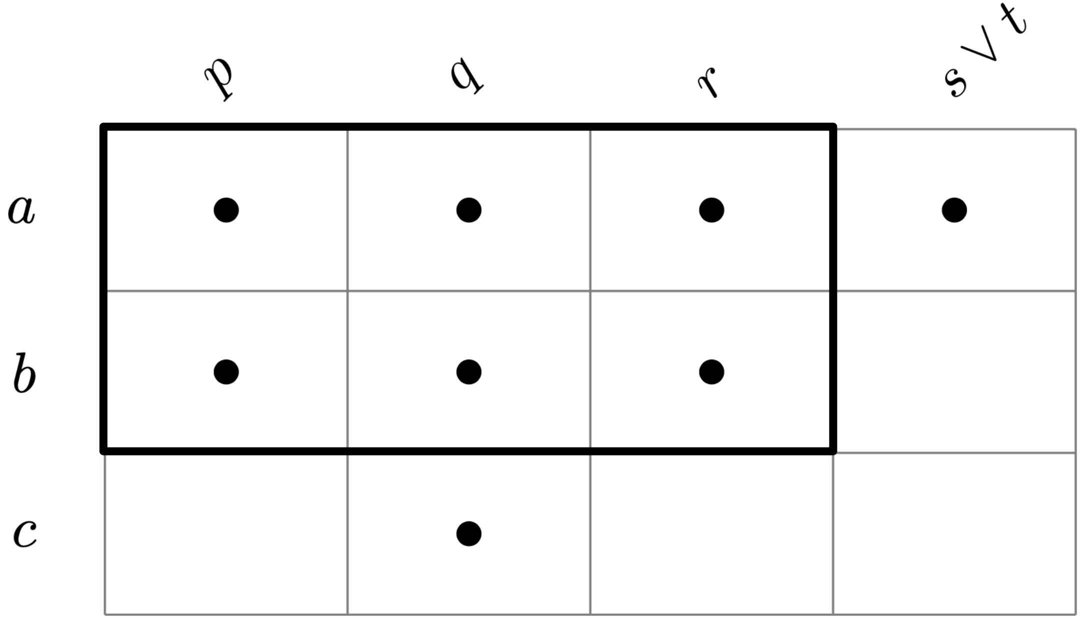
BVA finds grids of clauses in an iterative, greedy manner



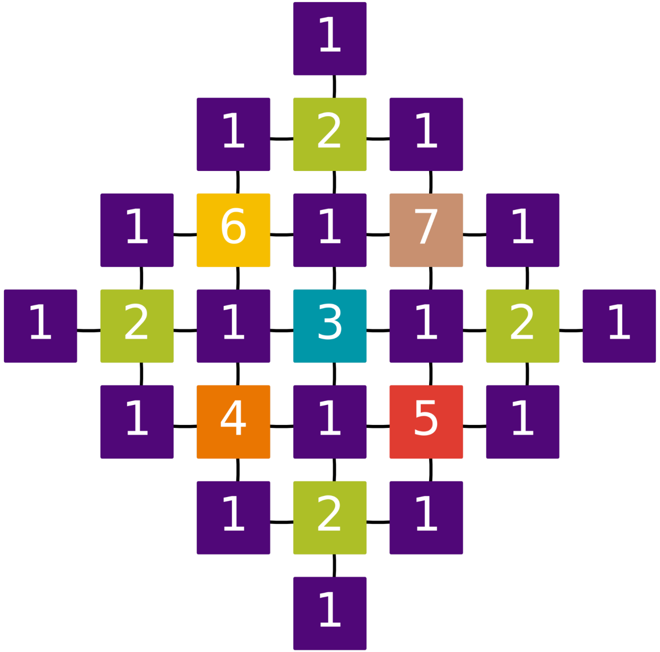
BVA finds grids of clauses in an iterative, greedy manner

	p	q	r	$s \vee t$
a	●	●	●	●
b	●	●	●	
c		●		

BVA finds grids of clauses in an iterative, greedy manner

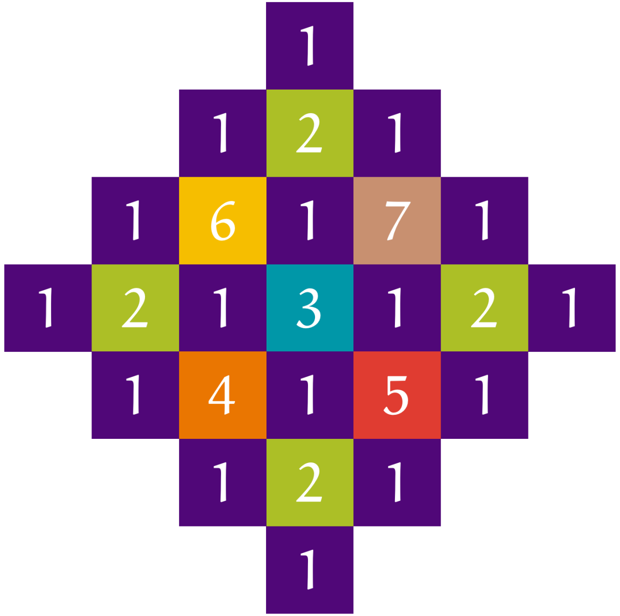


Example: The packing k-coloring of the square grid



Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

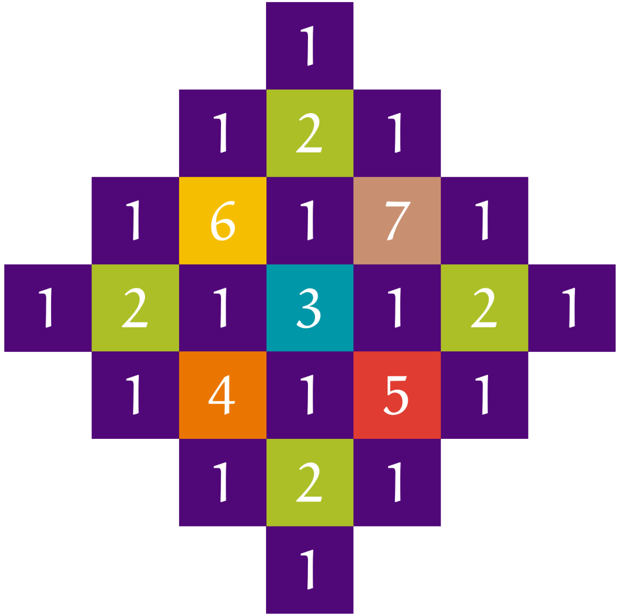
Example: The packing k-coloring of the square grid



Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

... ...

Example: The packing k-coloring of the square grid



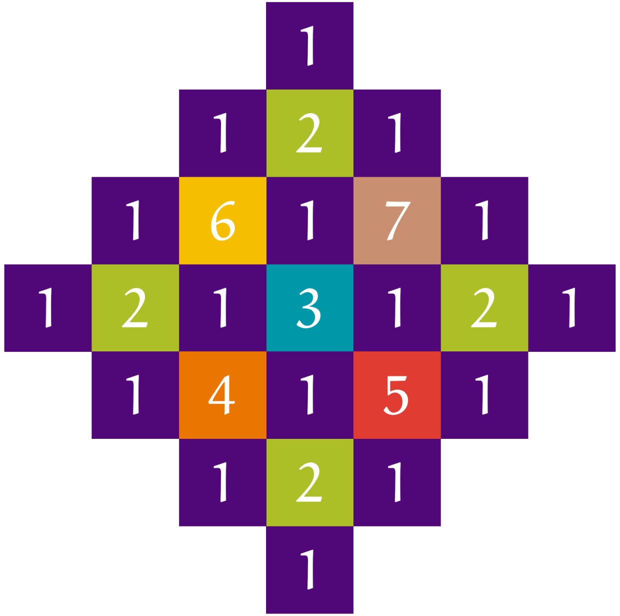
Direct encoding:

$v_{(x,y),k}$ denotes whether location (x, y) has color k

Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

... ...

Example: The packing k-coloring of the square grid



Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

Direct encoding:

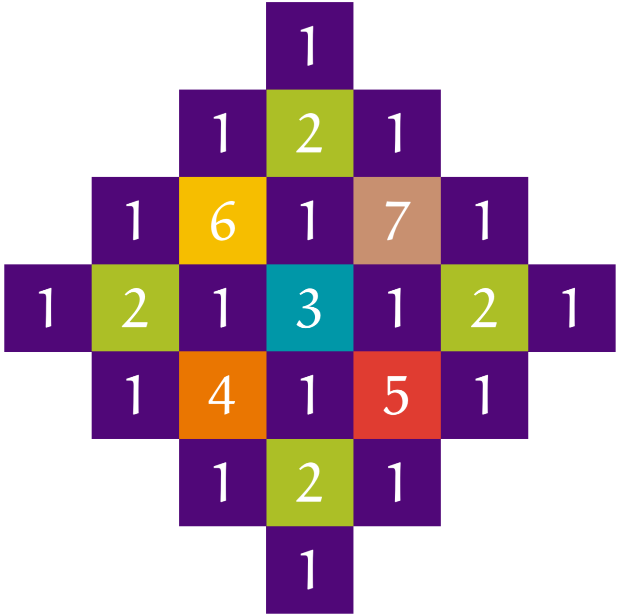
$v_{(x,y),k}$ denotes whether location (x, y) has color k

1. At-Least-One-Color clauses

$$v_{(0,0),1} \vee v_{(0,0),2} \vee \ldots \vee v_{(0,0),6}$$

… …

Example: The packing k-coloring of the square grid



Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

Direct encoding:

$v_{(x,y),k}$ denotes whether location (x, y) has color k

1. At-Least-One-Color clauses

$$v_{(0,0),1} \vee v_{(0,0),2} \vee \ldots \vee v_{(0,0),6}$$

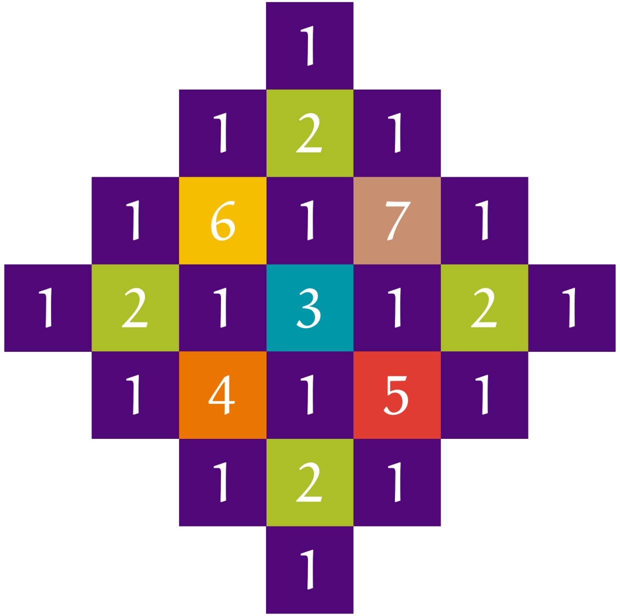
2. At-Most-One-Distance clauses

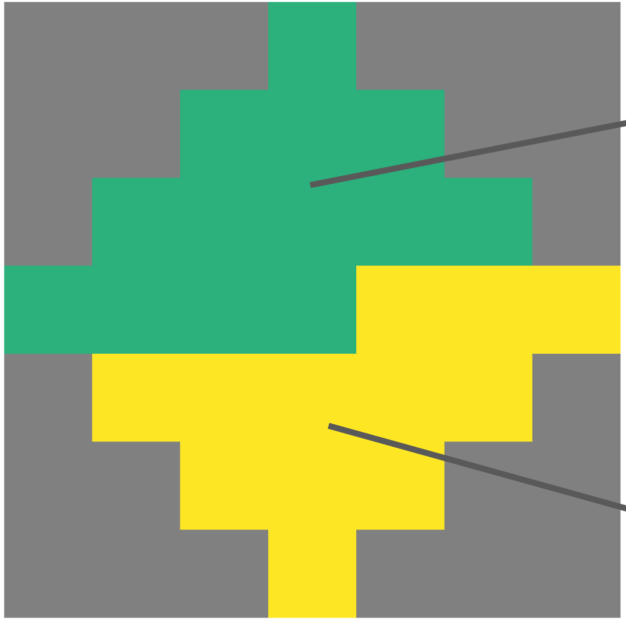
$$\overline{v_{(0,0),1}} \vee \overline{v_{(0,1),1}} \qquad \overline{v_{(0,0),2}} \vee \overline{v_{(0,1),2}}$$

$$\overline{v_{(0,0),1}} \vee \overline{v_{(1,0),1}} \qquad \overline{v_{(1,0),2}} \vee \overline{v_{(0,0),2}}$$

$$\ldots \qquad \ldots$$

Example: Auxiliary variables describe *regions* of variables



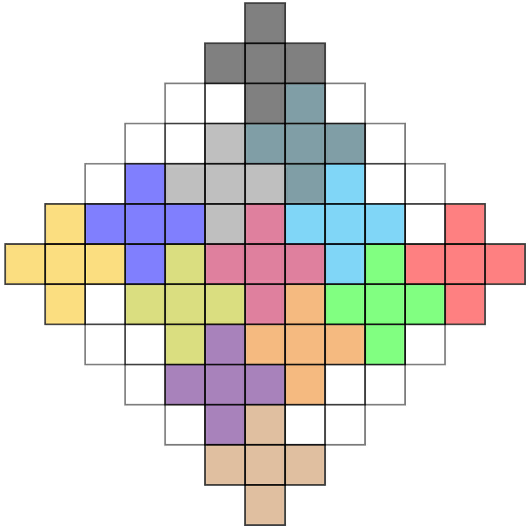


$$\boldsymbol{x} \vee \overline{v_{(3,0),6}}$$
$$\boldsymbol{x} \vee \overline{v_{(2,1),6}}$$
$$\boldsymbol{x} \vee \overline{v_{(2,2),6}}$$
$$\dots$$

$$\overline{\boldsymbol{x}} \vee \overline{v_{(4,3),6}}$$
$$\overline{\boldsymbol{x}} \vee \overline{v_{(4,4),6}}$$
$$\overline{\boldsymbol{x}} \vee \overline{v_{(4,5),6}}$$
$$\dots$$

Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

Example: Different ‘regions’ may be more effective



Visualization of the “plus” encoding

	direct encoding		bva encoding		plus encoding	
	$D_{5,10,5}$	$D_{6,11,6}$	$D_{5,10,5}$	$D_{6,11,6}$	$D_{5,10,5}$	$D_{6,11,6}$
Number of variables	610	935	973	1559	673	1039
Number of clauses	10688	21086	2313	3928	4063	7548
CDCL runtime (s)	255.12	10774.79	39.88	2539.38	15.90	811.66

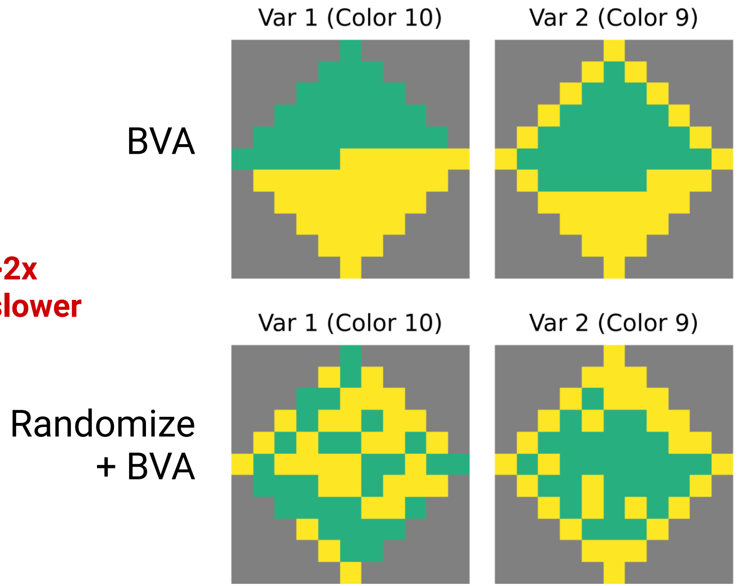
Subercaseaux and Heule (2023) studied variables added by BVA to derive an efficient manual encoding.

Subercaseaux, Bernardo, and Marijn JH Heule. "The packing chromatic number of the infinite square grid is 15." *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Cham: Springer Nature Switzerland, 2023.

BVA is sensitive to randomization

	Vars	Clauses	Time (s)
Original	610	10688	590
BVA	973	2313	**38**
Randomize + BVA	971	2305	**107**

>2x slower



Individual variable additions have a disproportionate effect

	Vars	Clauses	Time (s)
Original	610	10688	590
BVA	973	2313	38
Randomize + BVA	971	2305	107
1x BVA	611	9819	**105**
1x Randomize + BVA	611	9819	**429**

>5x faster

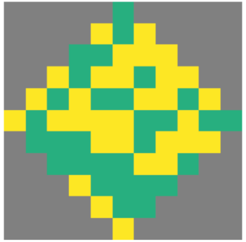
>4x slower

BVA

Var 1 (Color 10)

Randomize + BVA

Var 1 (Color 10)



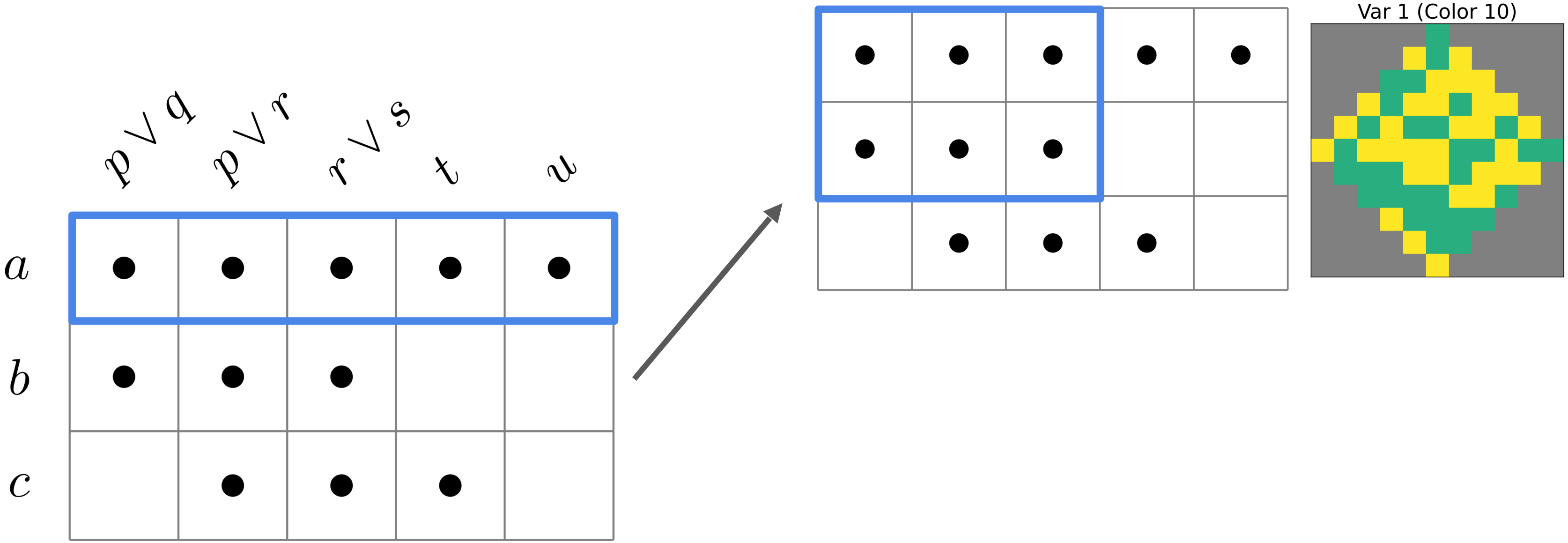
Tiebreaking in BVA leads to different auxiliary variables

	$p \vee q$	$p \vee r$	$r \vee s$	t	u
a	●	●	●	●	●
b	●	●	●		
c		●	●	●	

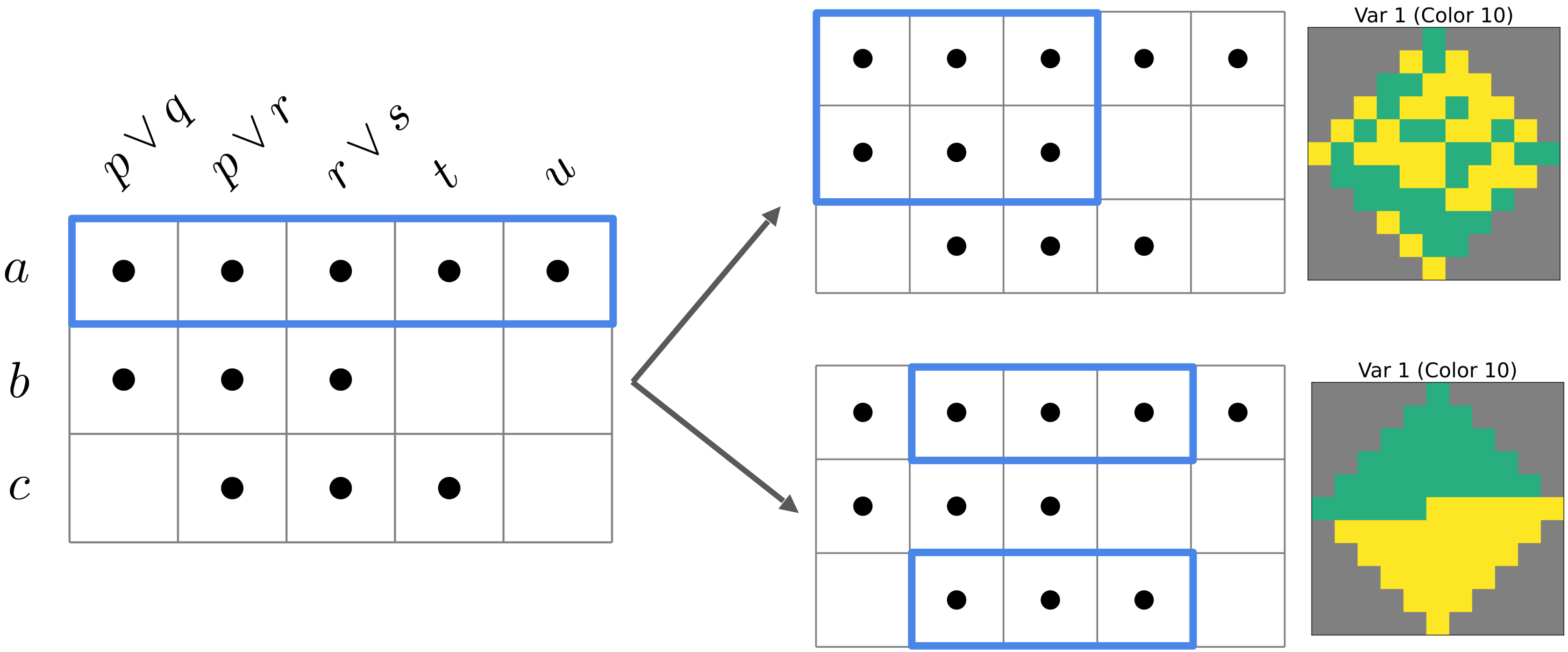
Tiebreaking in BVA leads to different auxiliary variables

	$p \lor q$	$p \lor r$	$r \lor s$	t	u
a	●	●	●	●	●
b	●	●	●		
c		●	●	●	

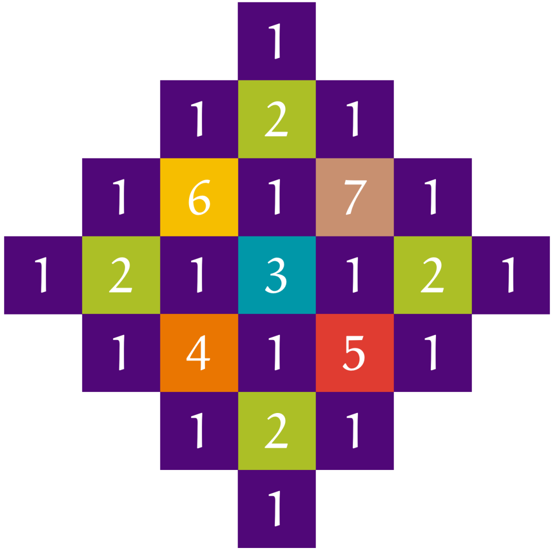
Tiebreaking in BVA leads to different auxiliary variables

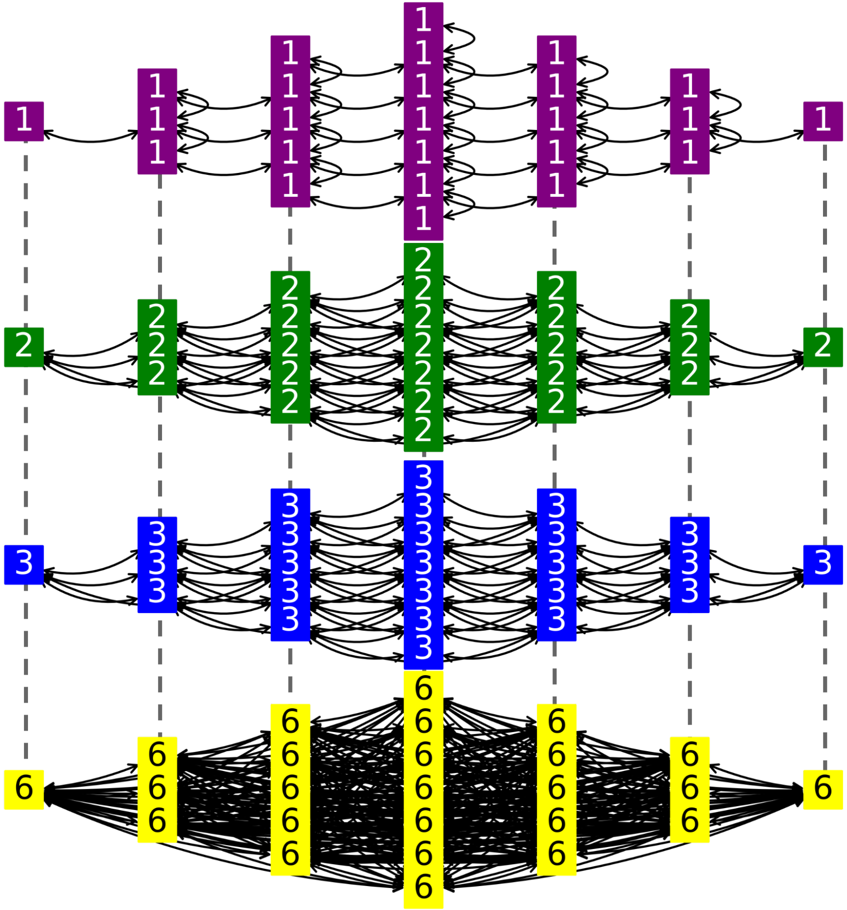


Tiebreaking in BVA leads to different auxiliary variables

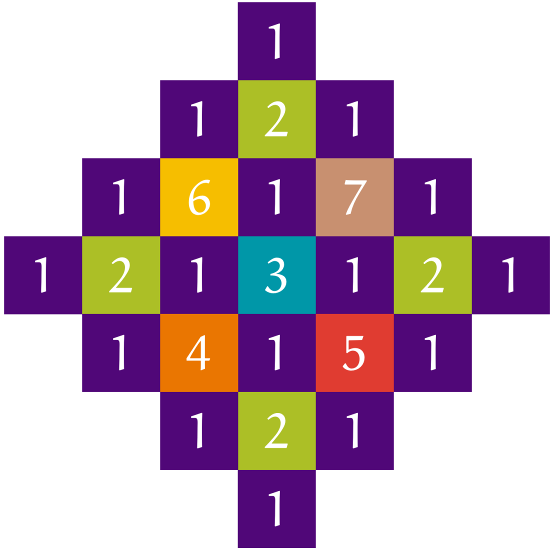


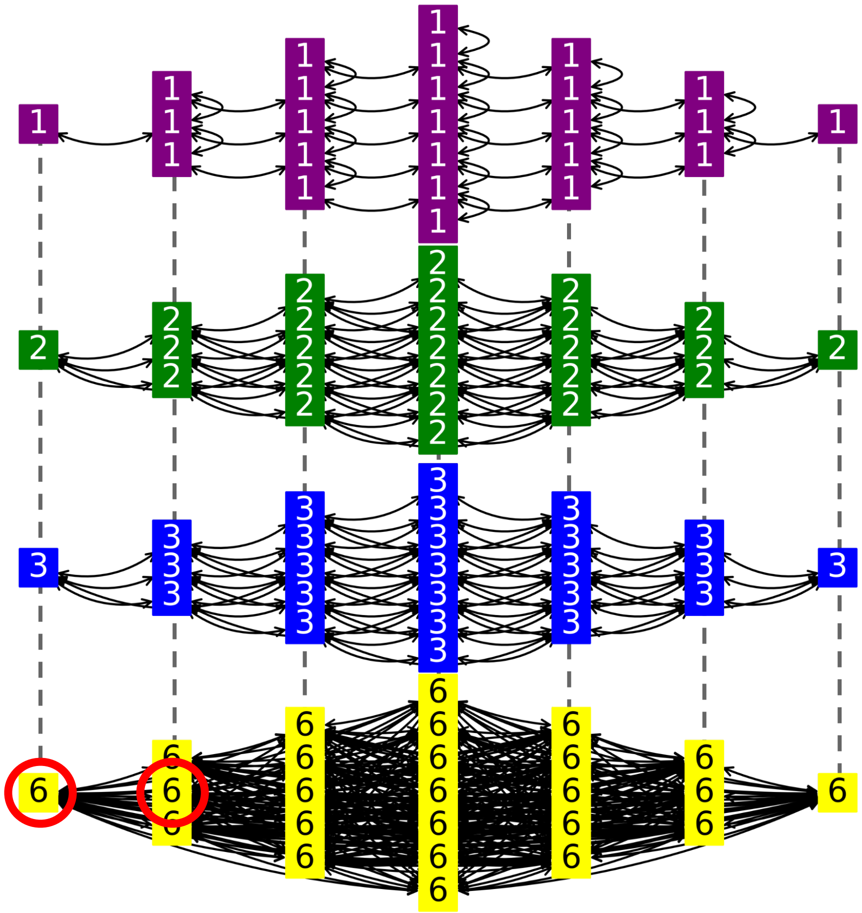
Using formula structure to break ties



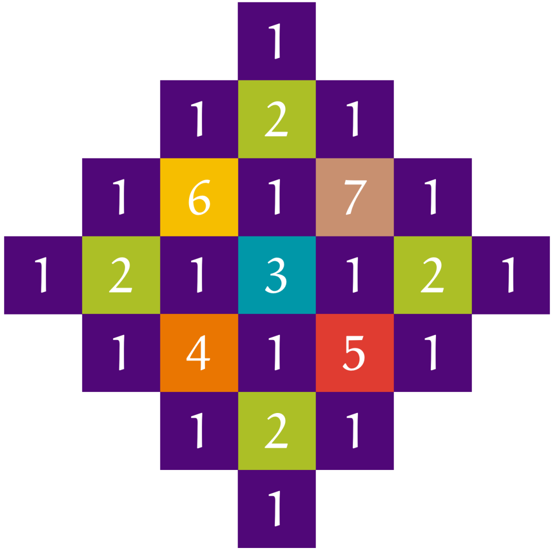


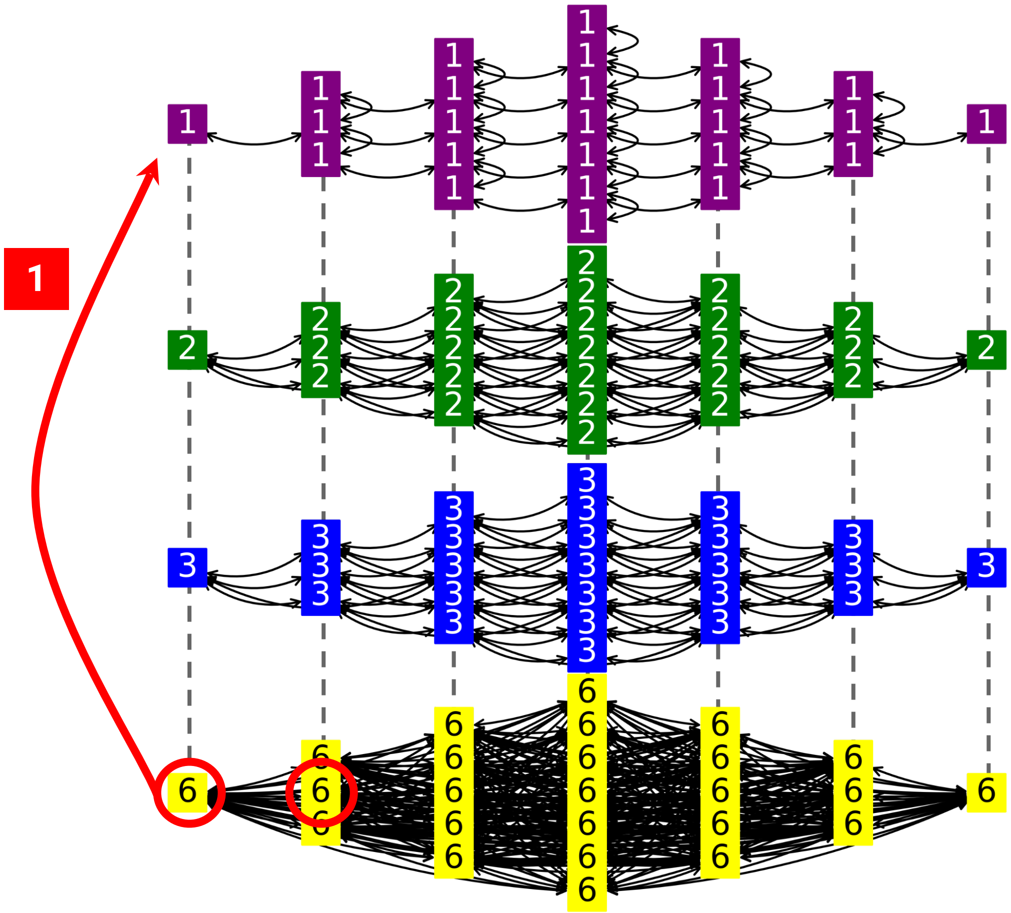
Using formula structure to break ties



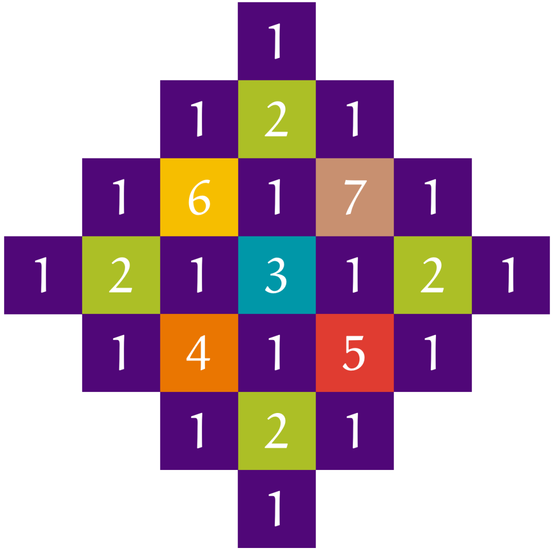


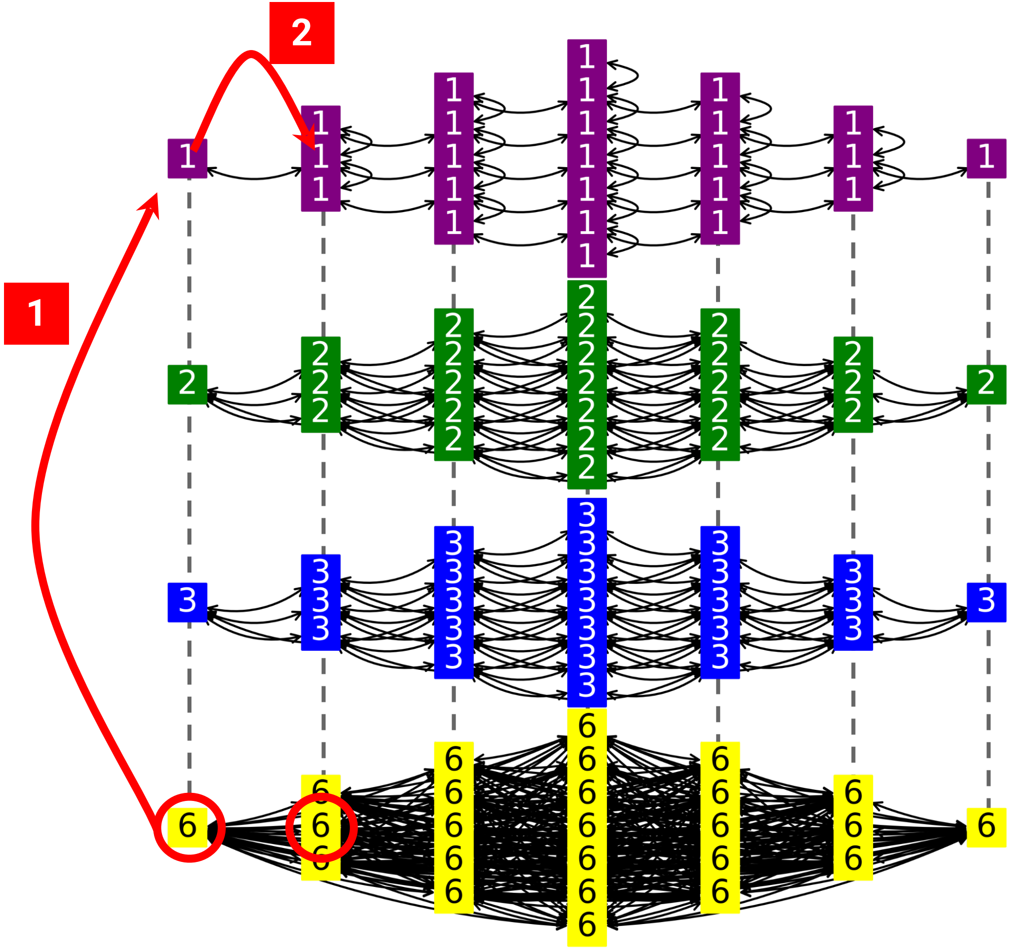
Using formula structure to break ties



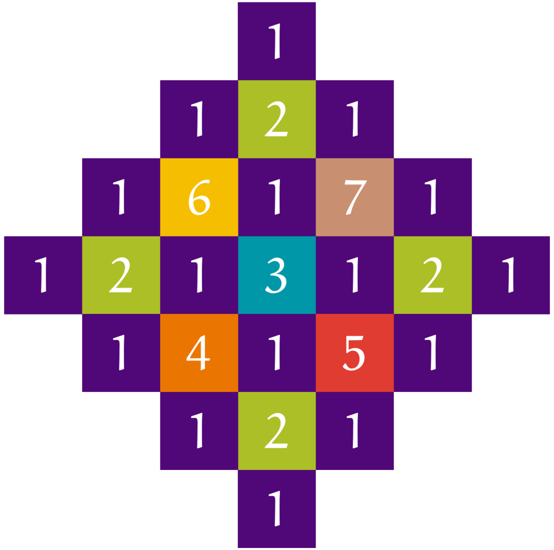


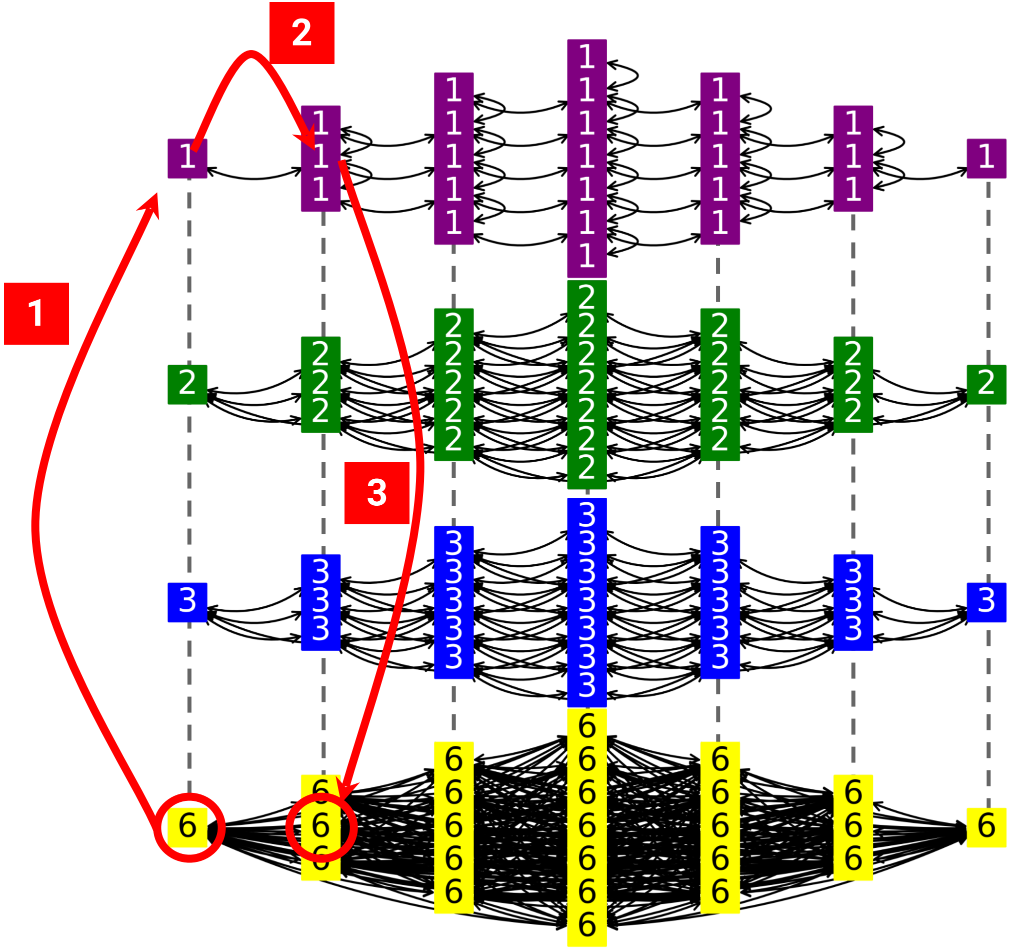
Using formula structure to break ties





Using formula structure to break ties





Breaking ties: Counting paths in the Variable Incidence Graph (VIG)

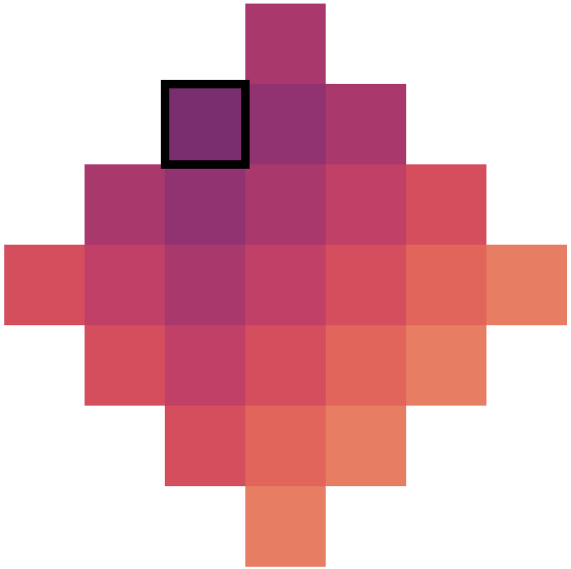
Variable incidence graph (VIG): a weighted graph of *variables*, where two variables share an edge for each clause between them

A: adjacency matrix of the VIG

H(x,y): number of paths of length 3 in the VIG

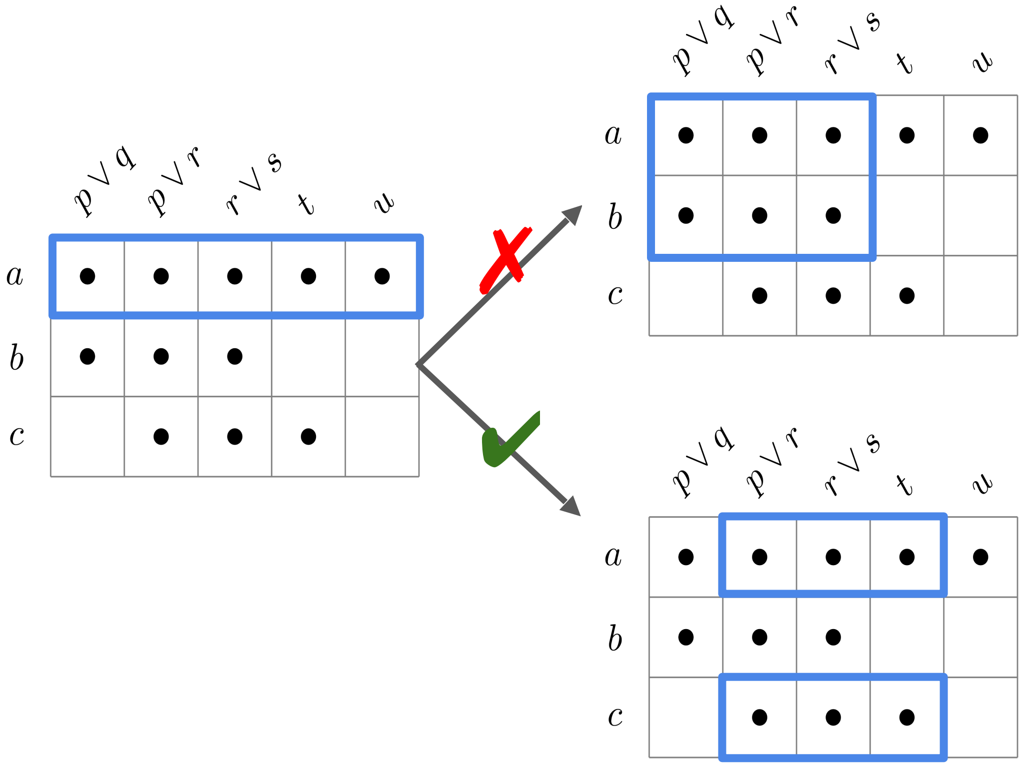
$$H(x,y) = (A^3)_{x,y}$$

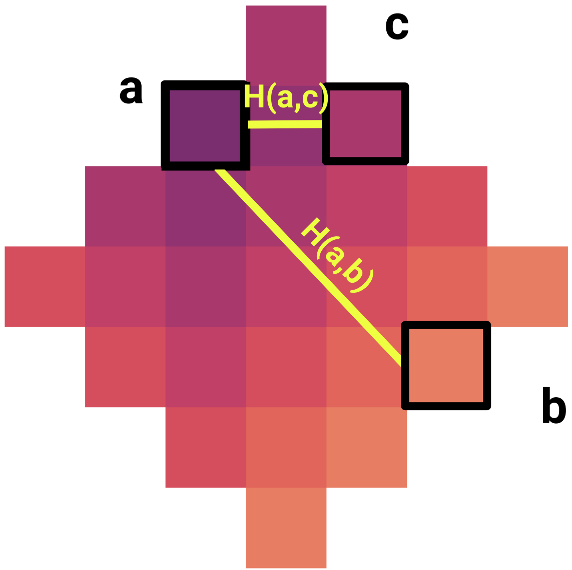
“Nearby” variables have **more paths of length 3 in the VIG** between them



H(x,y) evaluated between the highlighted cell and every other cell (for color 6 variables)

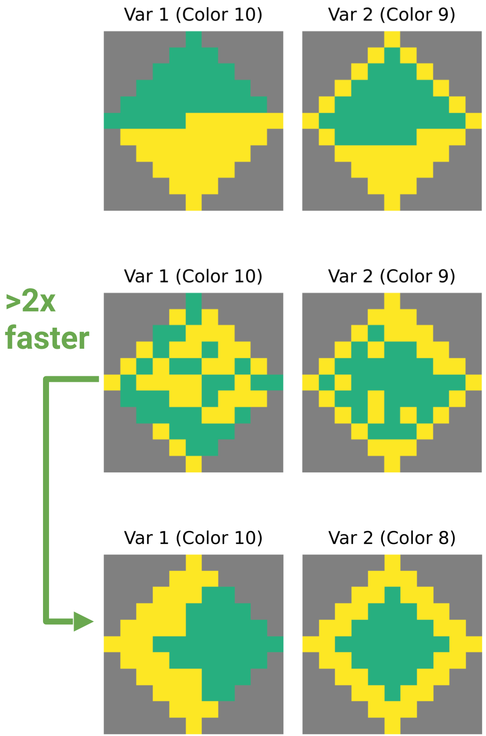
SBVA: Break ties using the "3-HOP" heuristic





Compare H(a, b) and H(a, c)

SBVA constructs effective auxiliary variables, even on randomized formulas



	Vars	Clauses	Time (s)
Original	610	10688	590
BVA	973	2313	38
Randomize + BVA	971	2305	**107**
Randomize + SBVA	972	2290	**55**

>2x faster

Evaluation Questions

Evaluation Questions

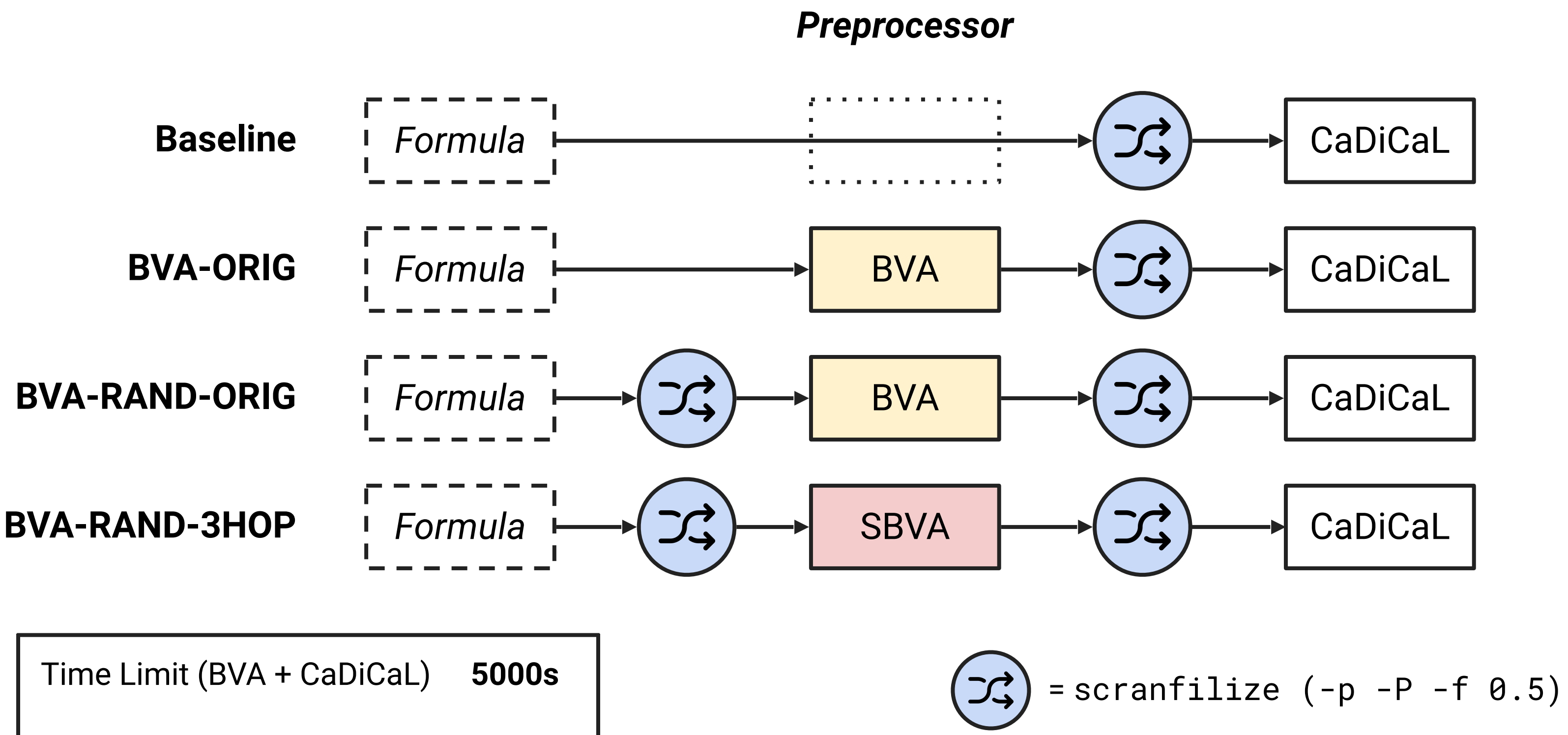
1. **Is SBVA worth running as a preprocessor?**
   a. Is it worth it to spend time preprocessing formulas before solving?
   b. How does SBVA compare to BVA (and to no preprocessor)?

Evaluation Questions

1. **Is SBVA worth running as a preprocessor?**
   a. Is it worth it to spend time preprocessing formulas before solving?
   b. How does SBVA compare to BVA (and to no preprocessor)?

2. **What types of formulas does SBVA work well on?**
   a. Are there common structures?
   b. Does reduction factor correlate with speedup?

Four Experimental Variants



Time Limit (BVA + CaDiCaL)	**5000s**
BVA Timeout	**200s**

Dataset

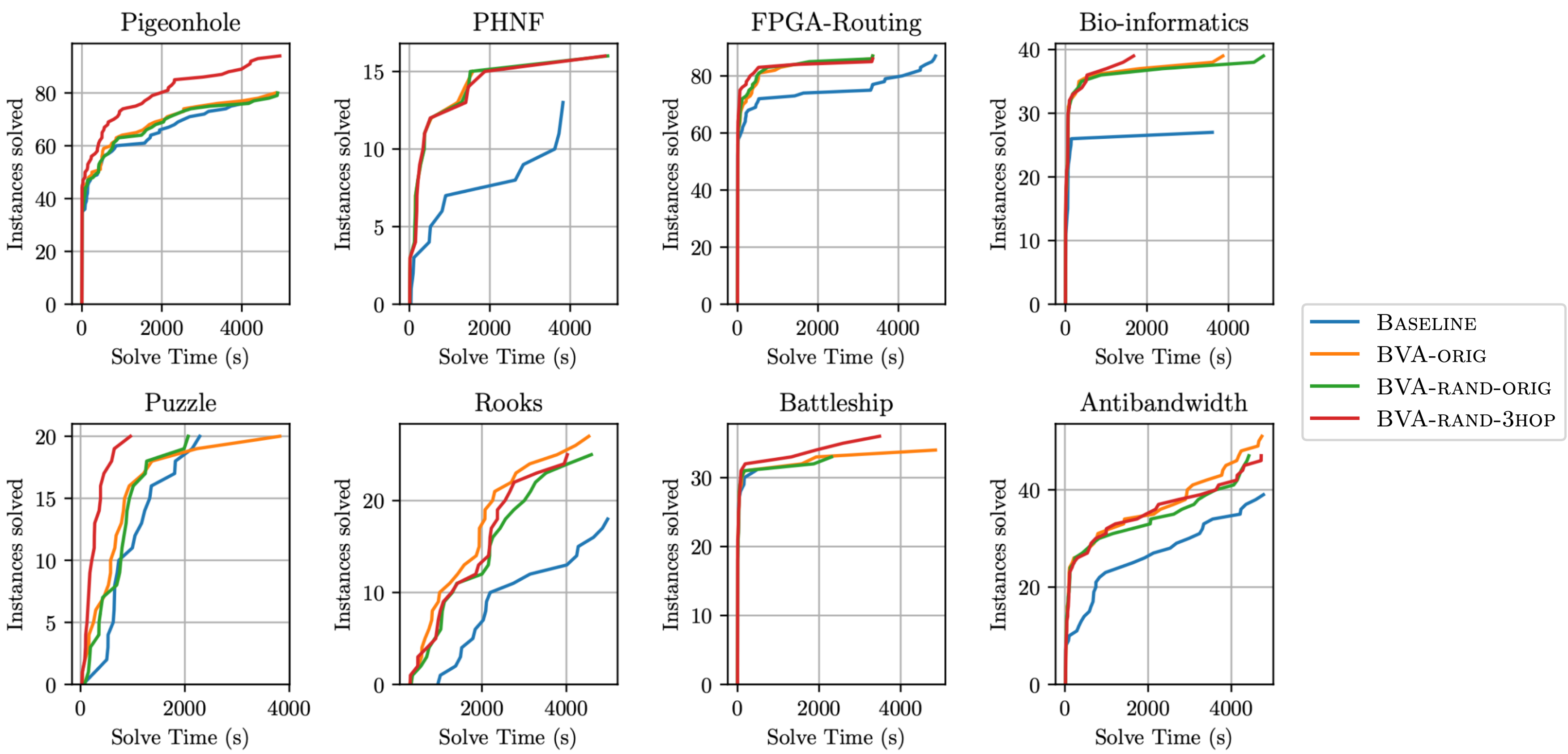
	Total	
Full	29,402	All formulas from the Global Benchmark Database[1], includes past SAT competitions
ANNI-2022	5,355	Anniversary Track from the 2022 SAT competition

[1]https://benchmark-database.de/

SBVA solved the most formulas in both datasets

Dataset	Variant	ALL		UNSAT		SAT	
		PAR-2	#	PAR-2	#	PAR-2	#
FULL	BASELINE	1077.91	21602	756.14	6495	1196.99	15107
	BVA-ORIG	867.04	22140	**635.71**	6562	948.85	15578
	BVA-RAND-ORIG	870.20	22077	673.58	6533	953.25	15544
	BVA-RAND-3HOP	**862.29**	**22173**	650.41	**6568**	**935.38**	**15605**
ANNI-2022	BASELINE	1262.18	3953	1164.61	2048	**1309.41**	1905
	BVA-ORIG	**1174.80**	3987	**967.85**	2085	1338.31	1902
	BVA-RAND-ORIG	1193.27	3958	1053.75	2060	1350.09	1898
	BVA-RAND-3HOP	1188.63	**3995**	982.84	**2088**	1350.98	**1907**

All times include BVA runtime


Pigeonhole
PHNF
FPGA-Routing
Bio-informatics
Puzzle
Rooks
Battleship
Antibandwidth
Instances solved
Solve Time (s)
Baseline
BVA-orig
BVA-rand-orig
BVA-rand-3hop

Pigeonhole Problems

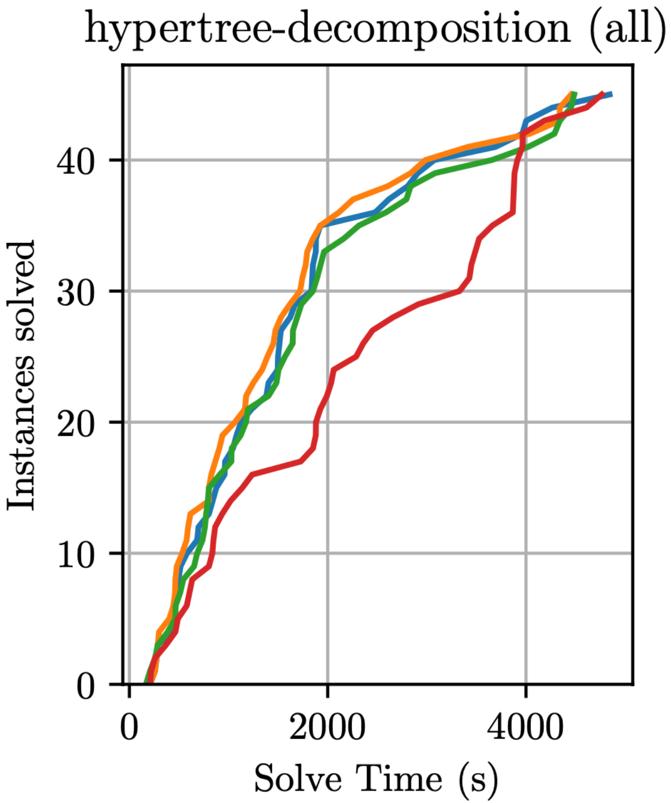
Total Solve Time (BVA + CaDiCaL) in seconds, T.O. = 5000

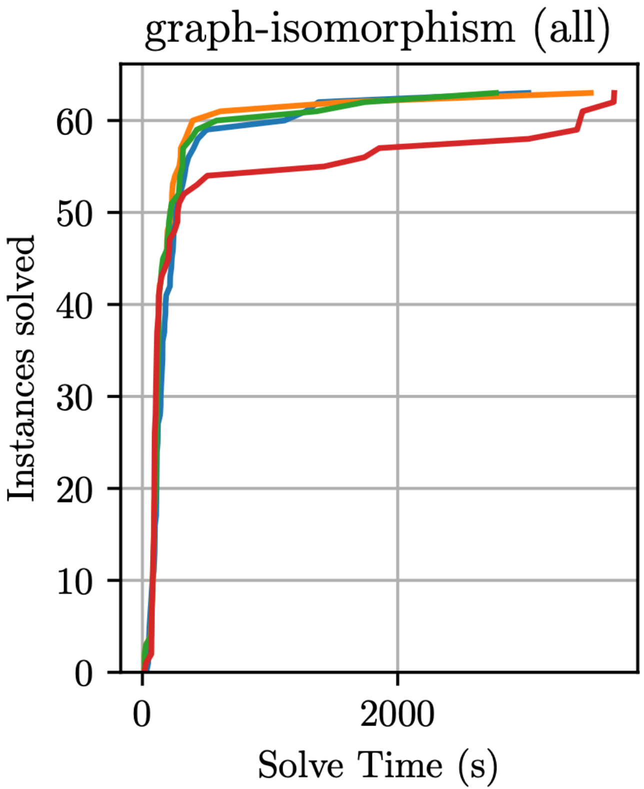
	Baseline	BVA-ORIG	BVA-RAND-ORIG	BVA-RAND-3HOP
fphp-010-008.shuffled-as.sat05-1213	0.5	0.2	0.3	**0.0**
fphp-010-009.shuffled-as.sat05-1227	5.4	4.3	4.2	**0.0**
fphp-012-010.shuffled-as.sat05-1214	113.4	45.1	36.2	**0.1**
fphp-012-011.shuffled-as.sat05-1228	1722.9	1525.2	1844.1	**0.7**
fphp-014-012.shuffled-as.sat05-1215	**T.O.**	**T.O.**	**T.O.**	**1.9**
fphp-014-013.shuffled-as.sat05-1229	**T.O.**	**T.O.**	**T.O.**	**564.5**
fphp-016-013.shuffled-as.sat05-1216	**T.O.**	**T.O.**	**T.O.**	**383.9**
fphp-016-015.shuffled-as.sat05-1230	**T.O.**	**T.O.**	**T.O.**	**1027.5**
fphp-018-014.shuffled-as.sat05-1217	**T.O.**	**T.O.**	**T.O.**	**549.1**
fphp-020-015.shuffled-as.sat05-1218	**T.O.**	**T.O.**	**T.O.**	**944.7**
php-010-008.shuffled-as.sat05-1171	0.6	0.2	0.3	**0.0**
php-010-009.shuffled-as.sat05-1185	6.6	3.1	3.6	**0.0**
php-012-010.shuffled-as.sat05-1172	138.3	31.6	24.0	**3.0**
php-012-011.shuffled-as.sat05-1186	2696.0	1006.3	755.8	**26.9**
php-014-012.shuffled-as.sat05-1173	**T.O.**	**T.O.**	**T.O.**	**237.5**
php-016-013.shuffled-as.sat05-1174	**T.O.**	**T.O.**	**T.O.**	**3024.1**

SBVA solves >2,500x faster!

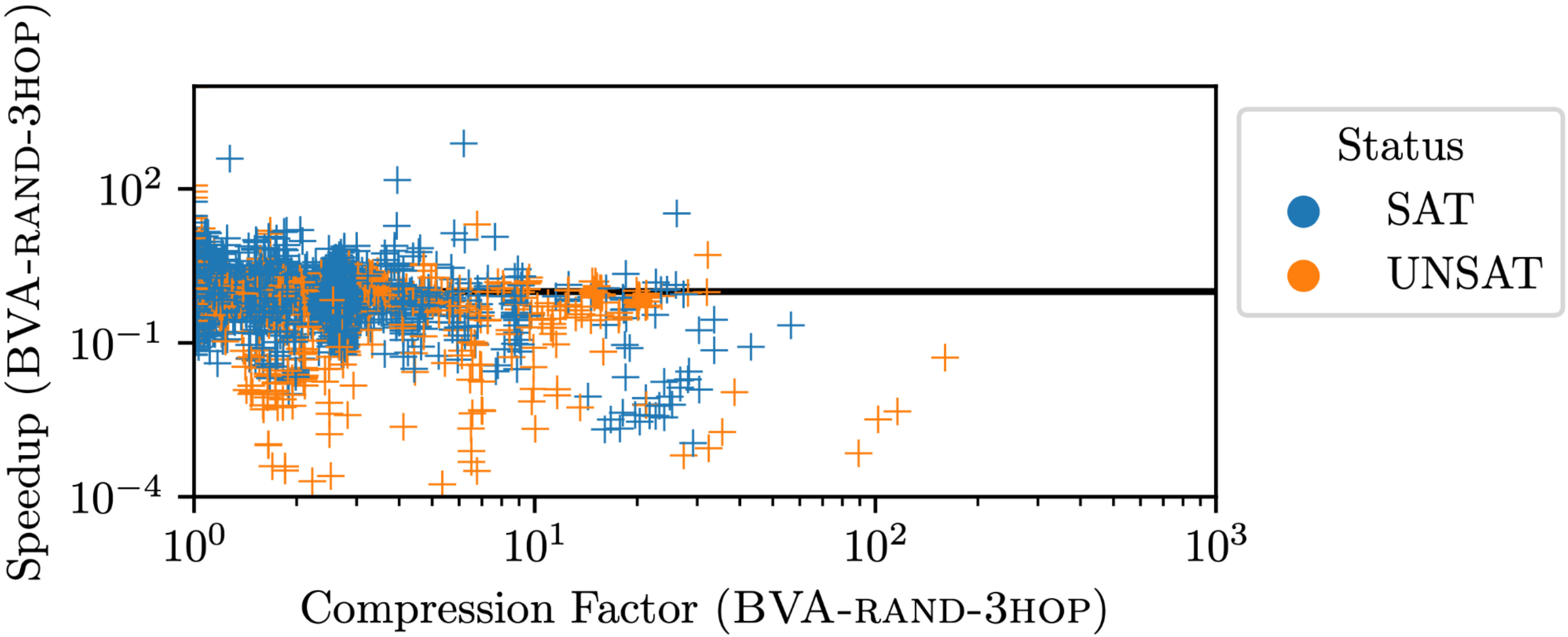
Puzzle

Total Solve Time (BVA + CaDiCaL) in seconds, T.O. = 5000				
	Baseline	BVA-ORIG	BVA-RAND-ORIG	BVA-RAND-3HOP
mp1-blockpuzzle_5x10_s2_free5	1326.3	159.4	352.9	**129.2**
mp1-blockpuzzle_5x10_s3_free4	1802.9	579.1	940.1	**360.0**
mp1-blockpuzzle_5x10_s5_free3	273.4	163.0	174.8	**85.5**
mp1-blockpuzzle_5x10_s7_free4	703.5	249.5	356.0	**142.2**
mp1-blockpuzzle_5x10_s8_free3	657.2	444.9	391.1	**158.0**
mp1-blockpuzzle_5x12_s6_free3	734.1	662.1	896.3	**275.5**
mp1-blockpuzzle_7x10_s10_free4	627.4	687.8	1017.4	**232.1**
mp1-blockpuzzle_7x10_s9_free6	527.2	89.0	156.0	**53.6**
mp1-blockpuzzle_8x8_s1_free4	649.9	295.2	427.8	**191.9**
mp1-blockpuzzle_9x9_s4_free3	2283.8	939.9	1276.8	**680.3**
mp1-blockpuzzle_9x9_s5_free3	529.5	851.5	760.1	**199.0**
mp1-blockpuzzle_9x9_s8_free3	994.5	1214.0	881.9	**413.9**


hypertree-decomposition (all)
Instances solved
Solve Time (s)
0
10
20
30
40
0
2000
4000


graph-isomorphism (all)
Instances solved
Solve Time (s)
0
10
20
30
40
50
60
0
2000

Reducing size only loosely correlates with speedup



Open-source Tool: SBVA

https://github.com/hgarrereyn/SBVA

- MIT License
- C++
- Generates DRAT proofs

as a preprocessor

```
./sbva -i formula.cnf -o reduced.cnf
```

with a solver
(automatically fixes proof and model)

```
./sbva_wrapped <solver> <formula> <proof>
```

Summary

- Bounded Variable Addition is sensitive to randomization
- Structured BVA augments BVA with a heuristic to construct more effective variable additions
- Structured BVA is effective on a variety of formulas, and is beneficial to run as a preprocessor