Implementation of bit-vector variables in a constraint solver with an application to the generation of cryptographic S-boxes

Kellen Dye

Prerequisites

bit $\rightarrow$ "binary digit" $\rightarrow$ 0 or 1

bit-vector $\rightarrow$ an array of bits, e.g. 1000110

bitwise operation $\rightarrow$ e.g. XOR ($\oplus$)

Constraint programming

Constraint programming

Declarative $\rightarrow$ Describe the solution, not the process

Constraint programming

Expresses relationships between variables

Constraint programming

Relationships: e.g. $x \leq y$ or $x \neq y$

Sudoku

Relationships between grid squares

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Sudoku

Row relationship: different values

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Sudoku

Column relationship: different values

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

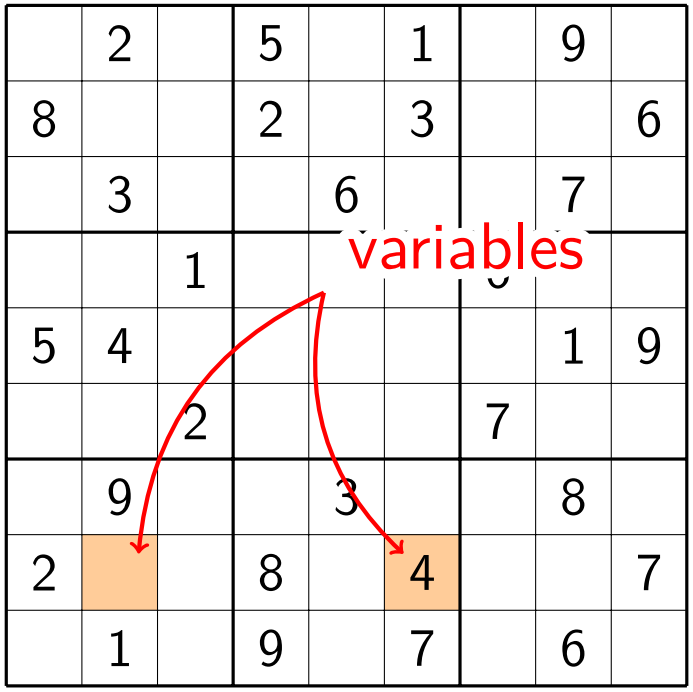
Sudoku

Block relationship: different values

	2		5		1		9	
8			2		3			6
	3			6			7	
		1				6		
5	4						1	9
		2				7		
	9			3			8	
2			8		4			7
	1		9		7		6	

Sudoku

Relationships between ~~grid squares~~ variables



Sudoku

In constraints

for i **in** $\{1..9\}$ **do**
    constrain alldifferent(row_i)
    constrain alldifferent$(column_i)$
    constrain alldifferent$(block_i)$

Constraint solvers

Reason on variables

Constraint solvers

Reason on ~~variables~~ variable domains

Constraint solvers

variable domains: e.g. "the values in $\{1, \ldots, 9\}$"

Propagators

Enforce constraints on variable domains

Propagators

Identify impossible values and remove them: e.g. $\{2, \ldots, 9\}$

Propagators

Can cause other propagators to be executed

Search

When no more propagation can occur

Search

e.g. two search branches: $x < 5$ and $x \geq 5$

Gecode (a constraint solver)

Support for integers, Booleans, floats, sets

Gecode (a constraint solver)

... but no bit-vectors

Bit-vector variables

Michel & Van Hentenryck

Bit-vector variables

Potential $O(1)$ constraints!

Bit-vector representation

$\langle lower, upper \rangle$

Bits same in both lower and upper are assigned/fixed.

Bit-vector variables

Implemented in Gecode

Bit-vector propagators

Implemented some defined by Michel & Van Hentenryck

Bit-vector propagators

New propagators for hamming weight, parity, disequality

Substitution boxes

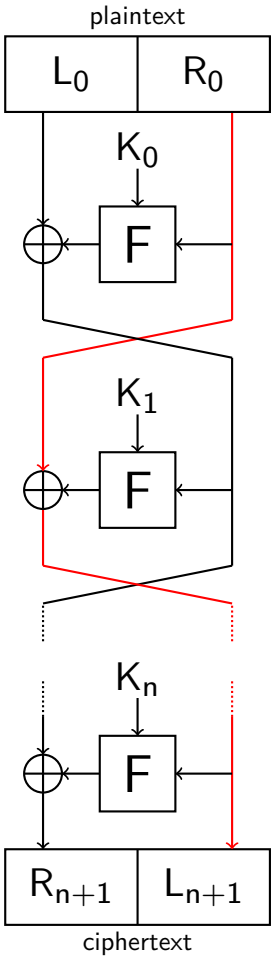
Substitution boxes

S_4		Middle 4 bits															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0111	1101	1110	0011	0000	0110	1001	1010	0001	0010	1000	0101	1011	1100	0100	1111
	01	1101	1000	1011	0101	0110	1111	0000	0011	0100	0111	0010	1100	0001	1010	1110	1001
	10	1010	0110	1001	0000	1100	1011	0111	1101	1111	0001	0011	1110	0101	0010	1000	0100
	11	0011	1111	0000	0110	1010	0001	1101	1000	1001	0100	0101	1011	1100	0111	0010	1110

$$S_4(110000) = 1111$$

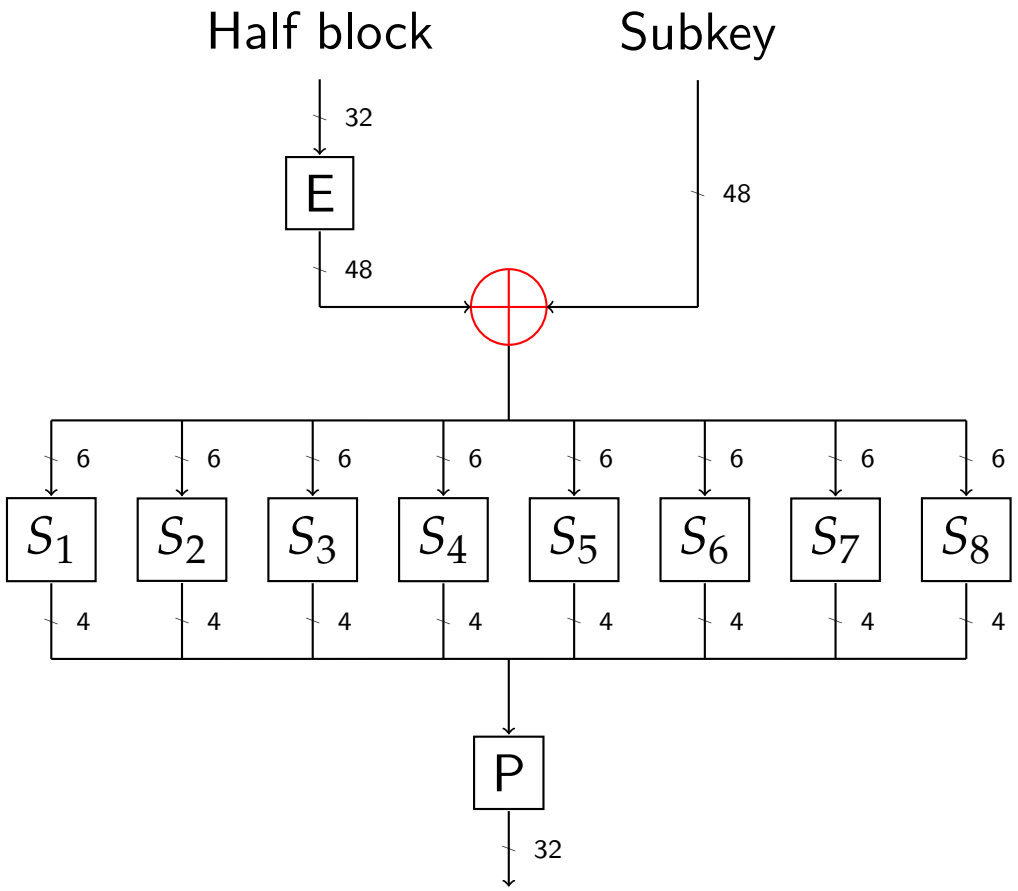
Substitution boxes

Feistel network



Substitution boxes

Feistel function F



Constraint programming + S-boxes

Suggested by Ramamoorthy et al.

"Good" substitution boxes

According to the DES design criteria

Relationships between values in the S-box:

- $x \neq y$
- weight$(S(x) \oplus S(y)) \geq 2$
- alldifferent(row_i)
- score$(S) \leq threshold$
- ...

S-box constraints

Implemented global propagators for DES design criteria S-2 and S-7

Symmetries

Reduce the search space

Symmetries

Ramamoorthy et al. describe several (rotation, "bit inversion")

Symmetries

New symmetries: reflective over both x and y axes

Comparison

Compare models based on set and Boolean variables

with several models based on bit-vectors

Models

Boolean, global integer-based S-2

set, global integer-based S-2

bit-vector, decomposed S-2 and S-7

bit-vector, global integer-based S-2, decomposed S-7

bit-vector, global bit-vector-based S-2, decomposed S-7

bit-vector, global bit-vector-based S-2, global bit-vector based S-7

Results

Bit-vector models are generally more effective

With global propagators $\rightarrow$ much more effective

Future work

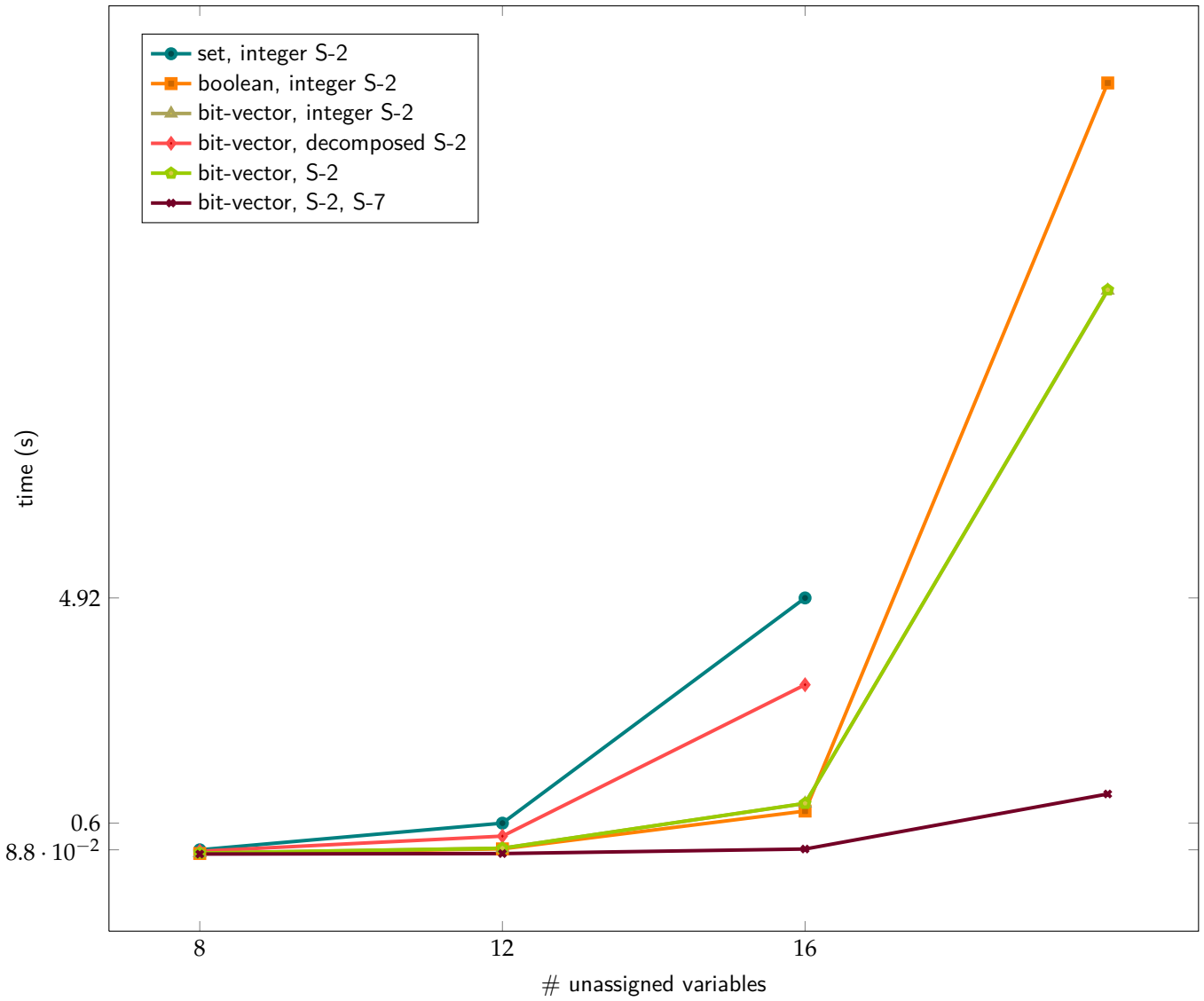
Implement additional propagators, integrate into Gecode core

Alternate requirements/constraints for S-boxes

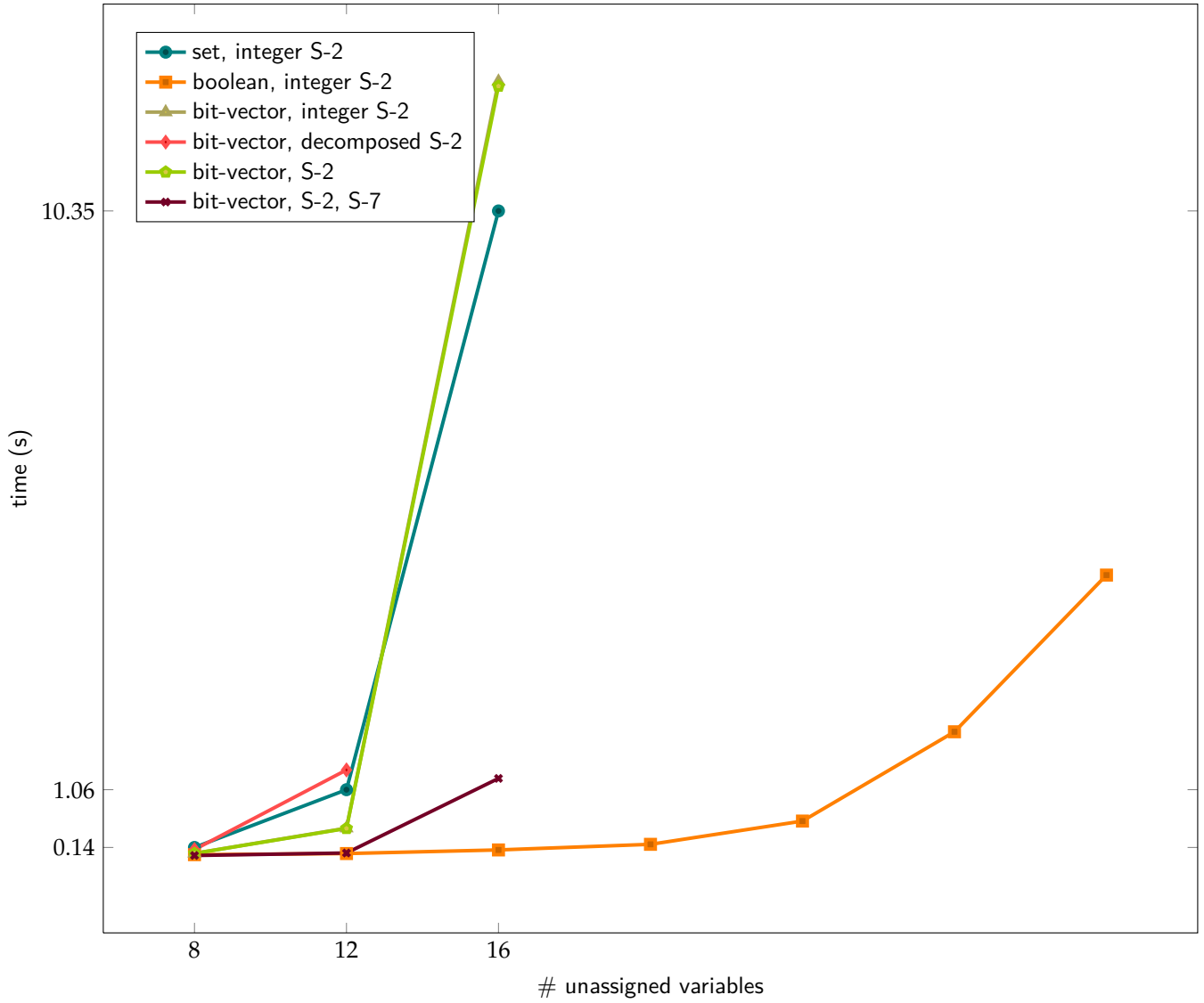
Thanks!

Graphs!

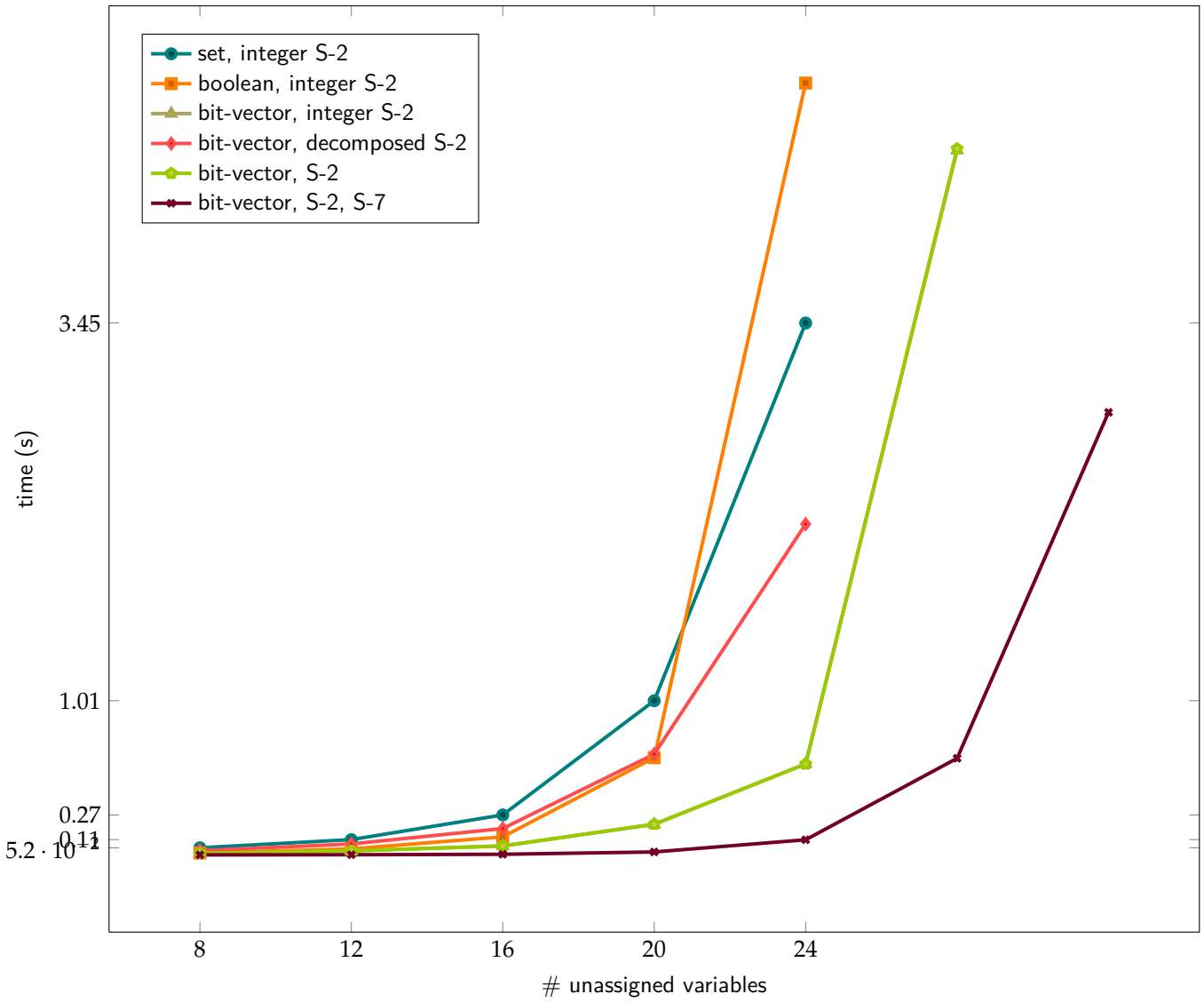
RND


set, integer S-2
boolean, integer S-2
bit-vector, integer S-2
bit-vector, decomposed S-2
bit-vector, S-2
bit-vector, S-2, S-7
time (s)
4.92
0.6
$8.8 \cdot 10^{-2}$
8
12
16
unassigned variables

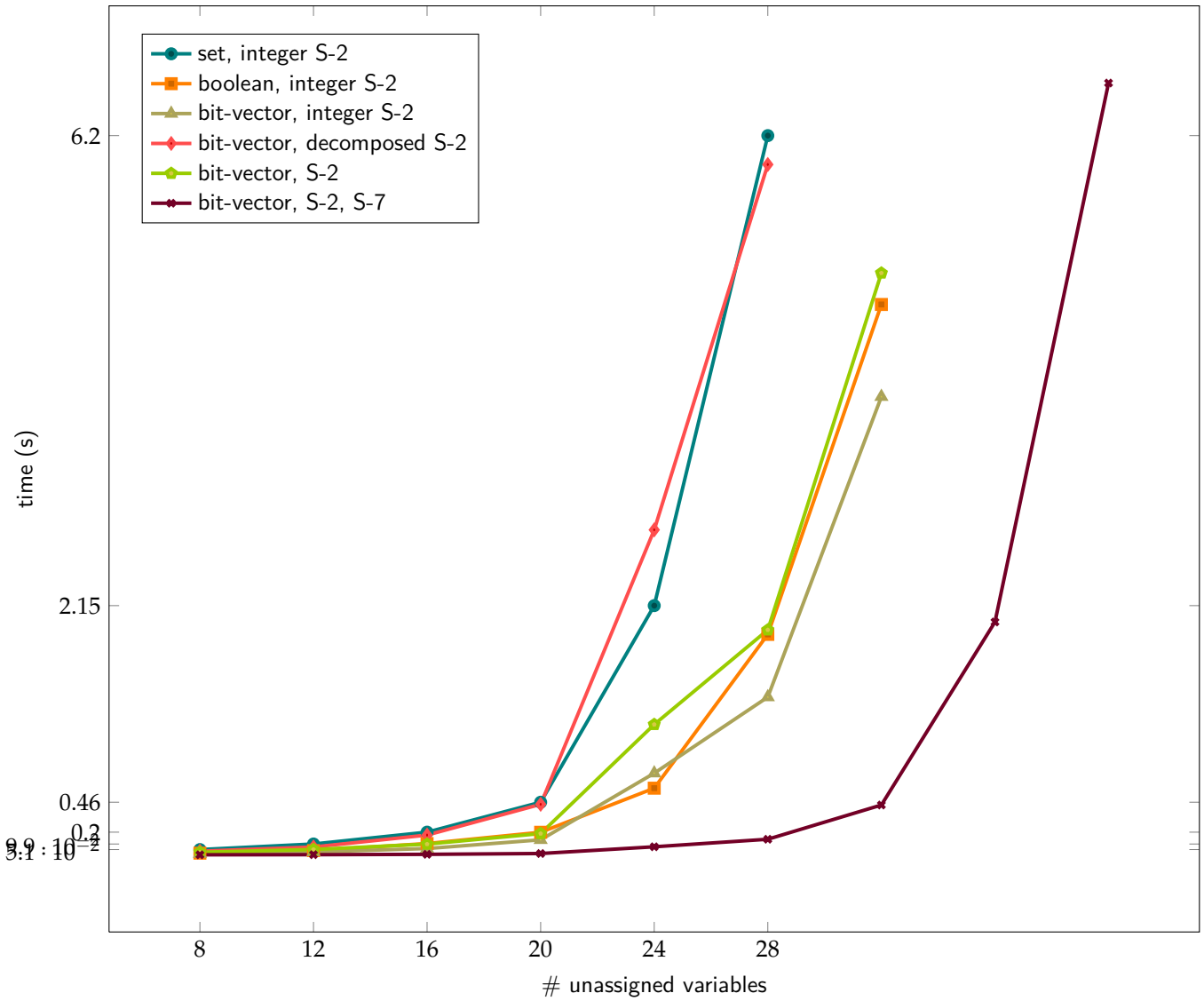
DEGREE


set, integer S-2
boolean, integer S-2
bit-vector, integer S-2
bit-vector, decomposed S-2
bit-vector, S-2
bit-vector, S-2, S-7
10.35
1.06
0.14
time (s)
8
12
16
unassigned variables

ACTIVITY


set, integer S-2
boolean, integer S-2
bit-vector, integer S-2
bit-vector, decomposed S-2
bit-vector, S-2
bit-vector, S-2, S-7
time (s)
3.45
1.01
0.27
0.11
5.2 · 10⁻²
8
12
16
20
24
unassigned variables

NONE


set, integer S-2
boolean, integer S-2
bit-vector, integer S-2
bit-vector, decomposed S-2
bit-vector, S-2
bit-vector, S-2, S-7
time (s)
6.2
2.15
0.46
0.2
8
12
16
20
24
28
unassigned variables