



UNIVERSITY OF AMSTERDAM

Analysing the correlation of symmetrical sudokus with their solving time using SAT solvers

Ruben Blom – 10684980

Luca Simonetto - 11413522

Symmetry in sudokus

The position of each given on the board is mirrored by another one.

- Horizontal (and vertical)
- Diagonal (and anti diagonal)
- Pi rotational
- Horizontal and vertical
- Diagonals

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

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

Hypothesis

Will the symmetry property of a sudoku puzzle affect the time required for a computer to solve it?

Dataset

- Datasets with symmetrical sudokus are very rare or non existent
- Each symmetry has to contain a reasonable number of sudokus
- All sudokus have to be labeled by difficulty



We created our own dataset with a custom generator

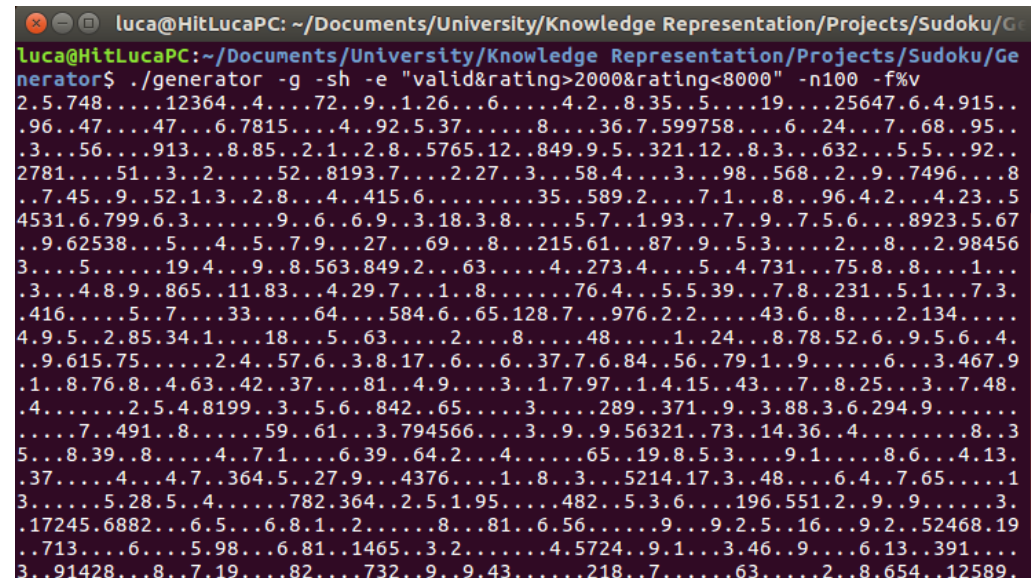
Generator

Created by Glenn Fowler

`./generator -g -sS -e "E" -nN -f%v -o O`

- -g generate sudokus
- -sS use S as symmetry (h, v, d, p, etc. for symmetries)
- -e “E” use E as generating expression (valid&rating>x&rating<y)
- -n N generate N sudokus
- -f%v remove unnecessary output
- -o O output in file O

huge amount of time needed to
generate the datasets!



```
luca@HitLucaPC: ~/Documents/University/Knowledge Representation/Projects/Sudoku/G
luca@HitLucaPC:~/Documents/University/Knowledge Representation/Projects/Sudoku/Ge
nerator$ ./generator -g -sh -e "valid&rating>2000&rating<8000" -n100 -f%v
2.5.748....12364..4....72..9..1.26...6.....4.2..8.35..5....19....25647.6.4.915..
.96..47....47...6.7815....4..92.5.37.....8....36.7.599758....6..24...7..68..95..
.3...56....913...8.85..2.1..2.8..5765.12..849.9.5..321.12..8.3...632...5.5...92..
2781...51..3..2....52..8193.7...2.27..3...58.4....3...98..568..2..9..7496...8
..7.45..9..52.1.3..2.8...4..415.6.....35..589.2....7.1...8...96.4.2...4.23..5
4531.6.799.6.3.....9..6..6.9..3.18.3.8....5.7..1.93...7..9..7.5.6...8923.5.67
..9.62538...5...4..5..7.9...27...69...8...215.61...87..9..5.3....2...8...2.98456
3....5.....19.4...9..8.563.849.2...63....4..273.4....5..4.731...75.8..8...1...
.3...4.8.9..865..11.83...4.29.7...1..8.....76.4...5.5.39...7.8..231..5.1...7.3.
.416....5..7...33.....64...584.6..65.128.7...976.2.2....43.6..8...2.134....
4.9.5..2.85.34.1...18...5..63...2...8...48...1..24...8.78.52.6..9.5.6..4.
..9.615.75....2.4..57.6..3.8.17..6...6..37.7.6.84..56..79.1..9.....6...3.467.9
.1..8.76.8..4.63..42..37...81..4.9...3..1.7.97..1.4.15..43...7..8.25...3..7.48.
.4.....2.5.4.8199..3..5.6..842..65....3.....289..371..9..3.88.3.6.294.9.....
.....7..491..8.....59..61...3.794566...3..9..9.56321..73..14.36..4.....8..3
5...8.39..8.....4..7.1...6.39..64.2...4.....65..19.8.5.3...9..1.....8.6...4.13.
.37....4...4.7..364.5..27.9...4376...1..8..3...5214.17.3..48...6.4..7.65....1
3.....5.28.5..4.....782.364..2.5.1.95....482..5.3.6....196.551.2..9..9.....3.
..17245.6882...6.5...6.8.1..2.....8...81..6.56...9..9.2.5..16...9.2..52468.19
..713...6...5.98...6.81..1465..3.2.....4.5724..9.1...3.46..9...6.13..391...
3..91428...8..7.19....82...732..9..9.43.....218..7.....63.....2..8.654..12589.
```

Difficulties

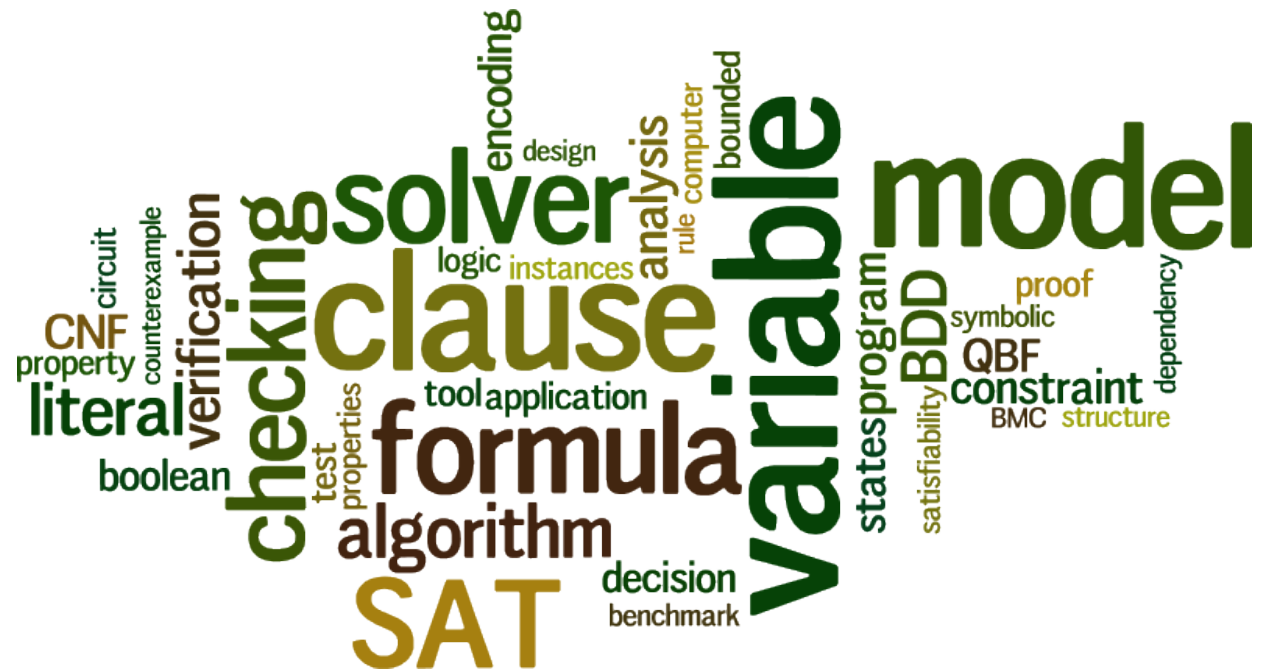
The generator has a difficulty rating ranging from 0 to 9999. We created five ranges in order to divide the dataset in five difficulties

Difficulty	Range
Easy	0-2000
Medium	2001-4000
Hard	4001-6000
Very hard	6001-8000
Extreme	8001-9999

SAT solver

We chose PycoSat, Python wrapper of PicoSat

- Deterministic
- Python ready
- Easy to use



Metric

The chosen metric is **ELAPSED TIME**

Why?

- Well fits our needs
- Easy to implement
- No need to reverse engineer PycoSat to get data

Problems

- Prone to misreading due to other running processes
- Fast computers may get reading that are too small to be precise and comparable

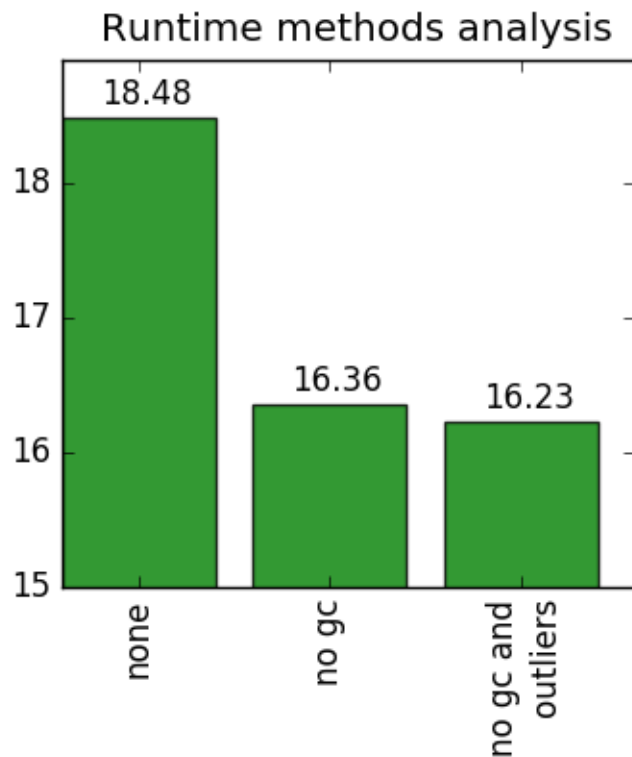
Solution use CPU time instead

- Hardware independent
- Pure reading
- Less susceptible to other processes

Metric

The garbage collector has been disabled when waiting for a solution from the SAT solver.

Why?

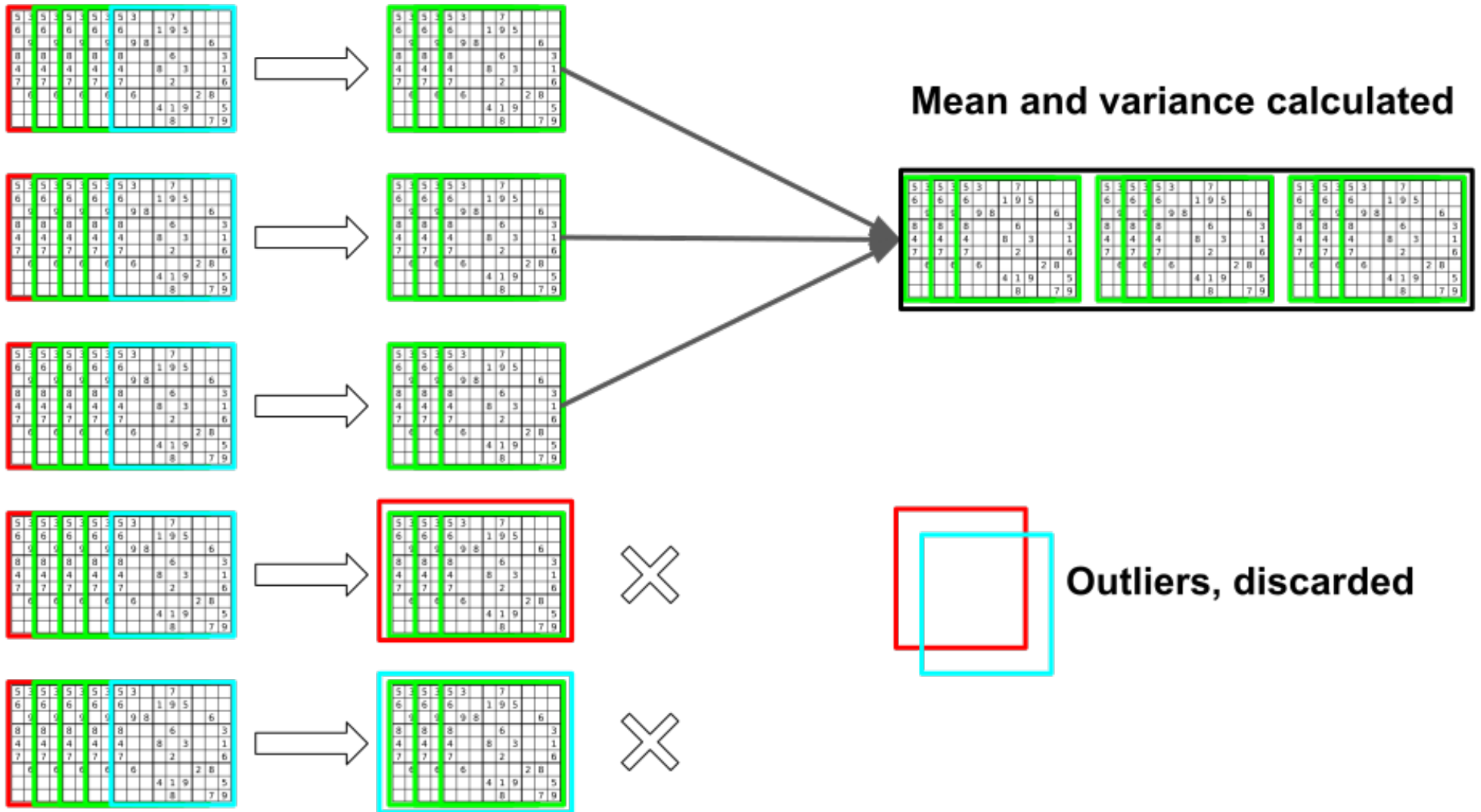


If the garbage collector is executed when running the SAT solver, the elapsed times increases by 12%!

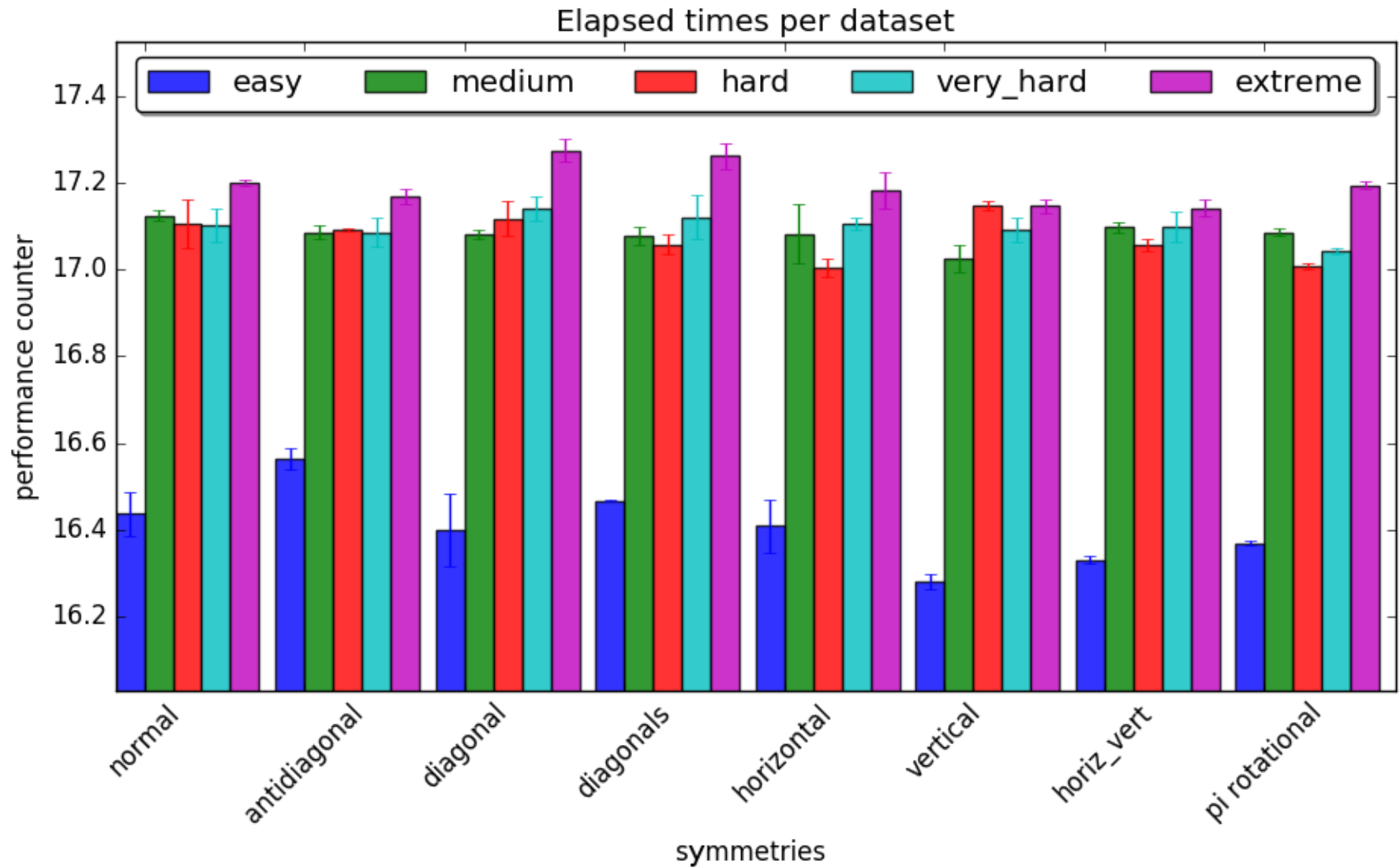
Experimental Procedure

- Run the SAT solver across each dataset N times
- After each run, remove the two sudoku outliers (short processes influence)
- After N runs, remove the two run outliers (medium processes influence)
- Compute mean and variance

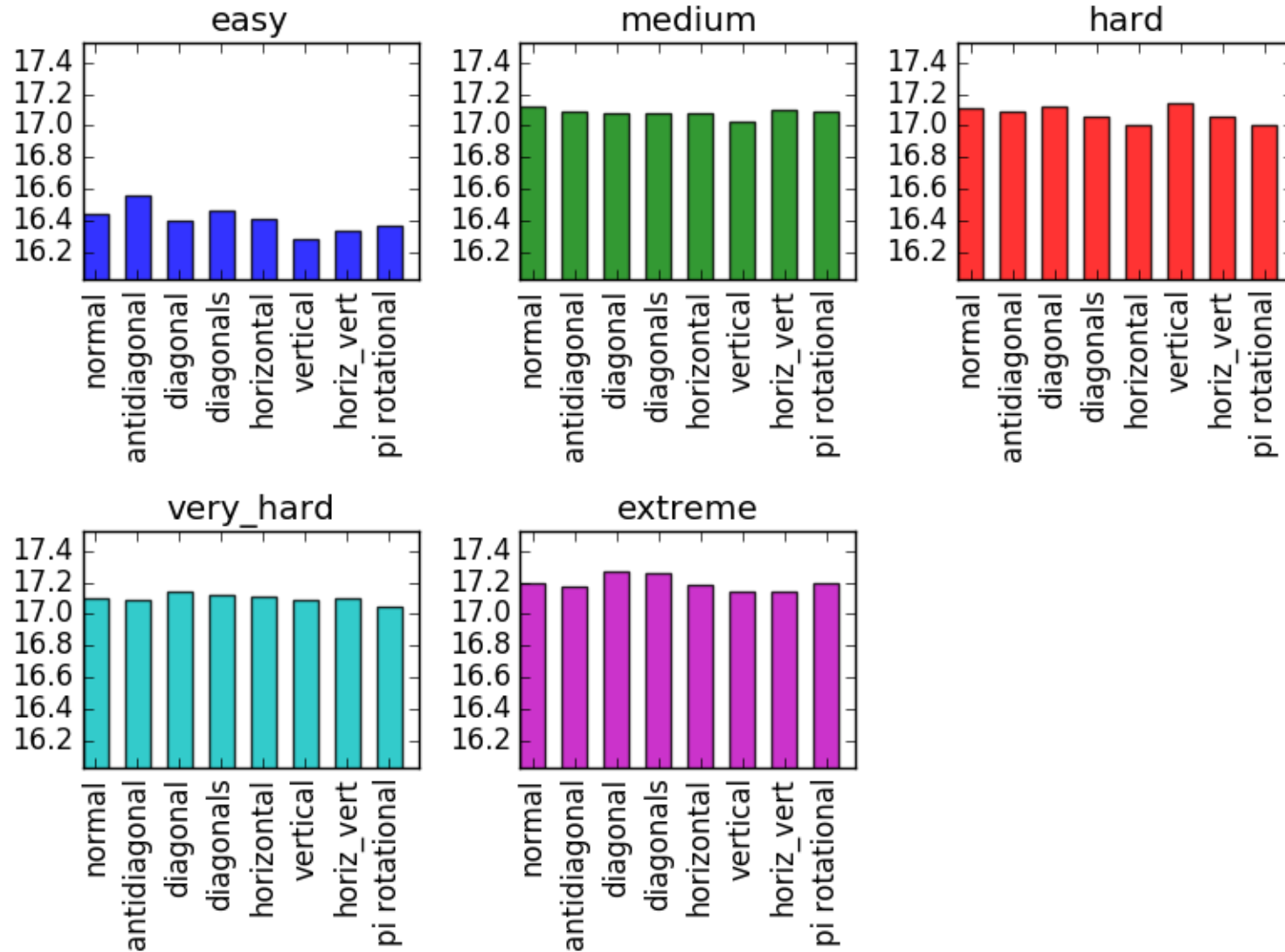
Experimental Procedure



Experimental results



Experimental results



Experimental results

- **The hypothesis is considered falsified**
- Easy sudokus take a noticeably lower time to be solved
- Visible differences are too small to be significant (around 0.3 milliseconds maximum)
- The variance in the results is very marginal (every time under 0.1)

Future work

- Use different metrics
- Compare different SAT solvers
- Use different techniques
- ...

Thank you!