UNIVERSITY OF AMSTERDAM

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

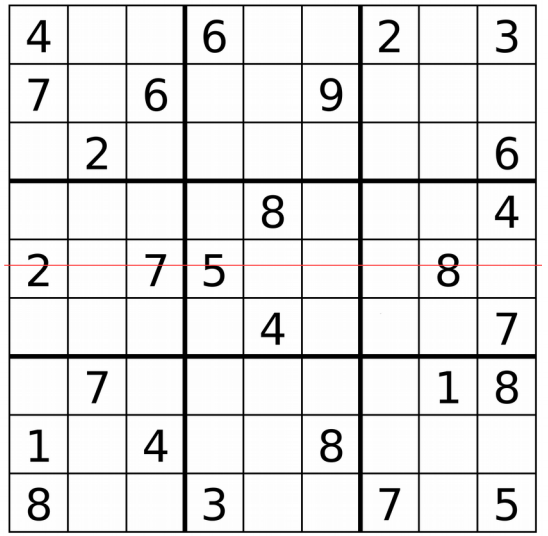
Ruben Blom – 10684980

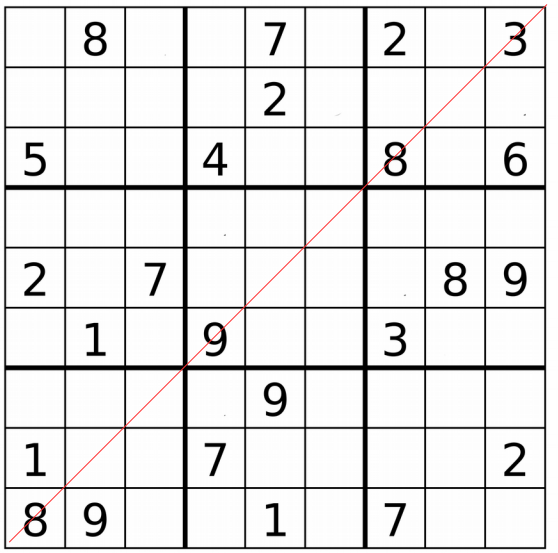
Luca Simonetto - 11413522

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





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

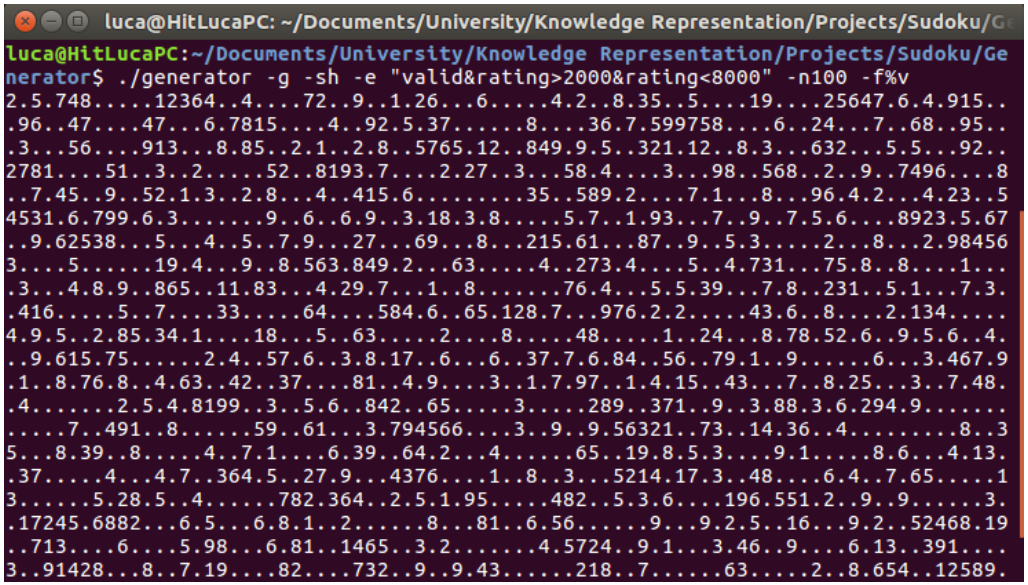
Generator

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!



Difficulties

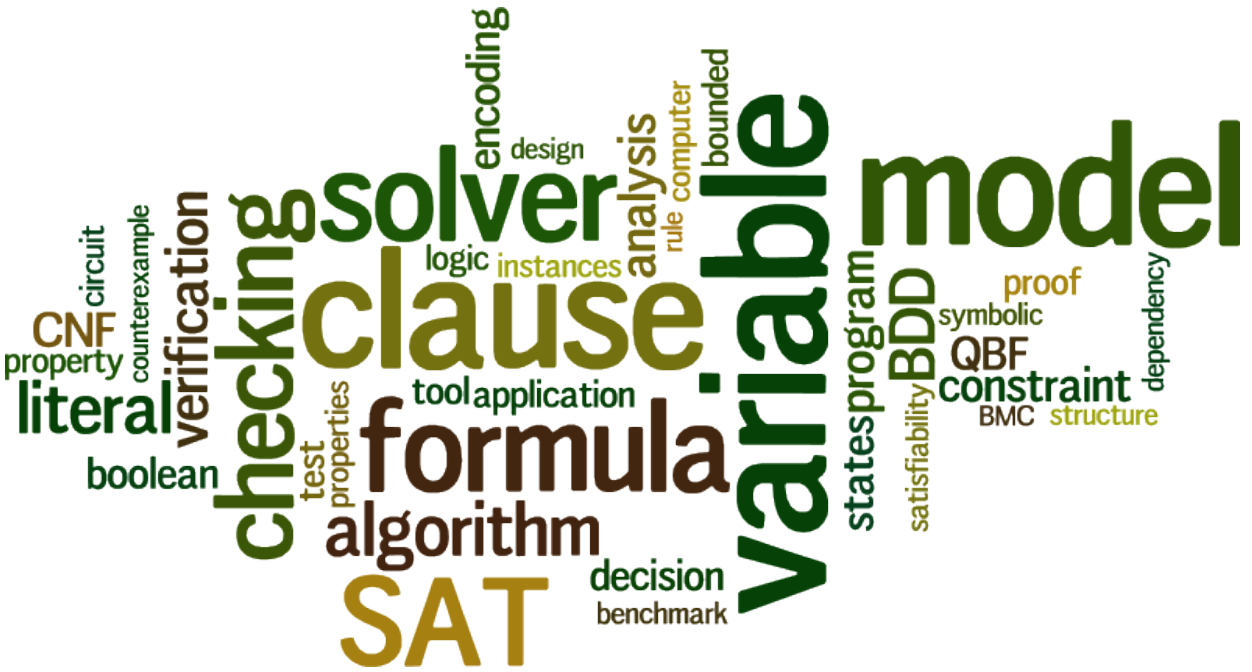
The generator has a difficulty rating ranging from 0 to 9999. We created five ranges in order to divide the dataset in five difficulties

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

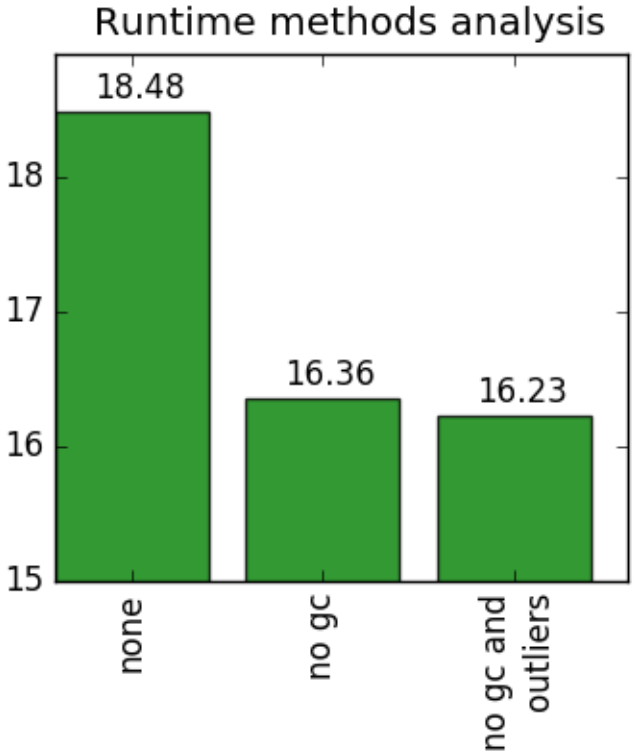
Solution use *CPU time* instead

- 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?

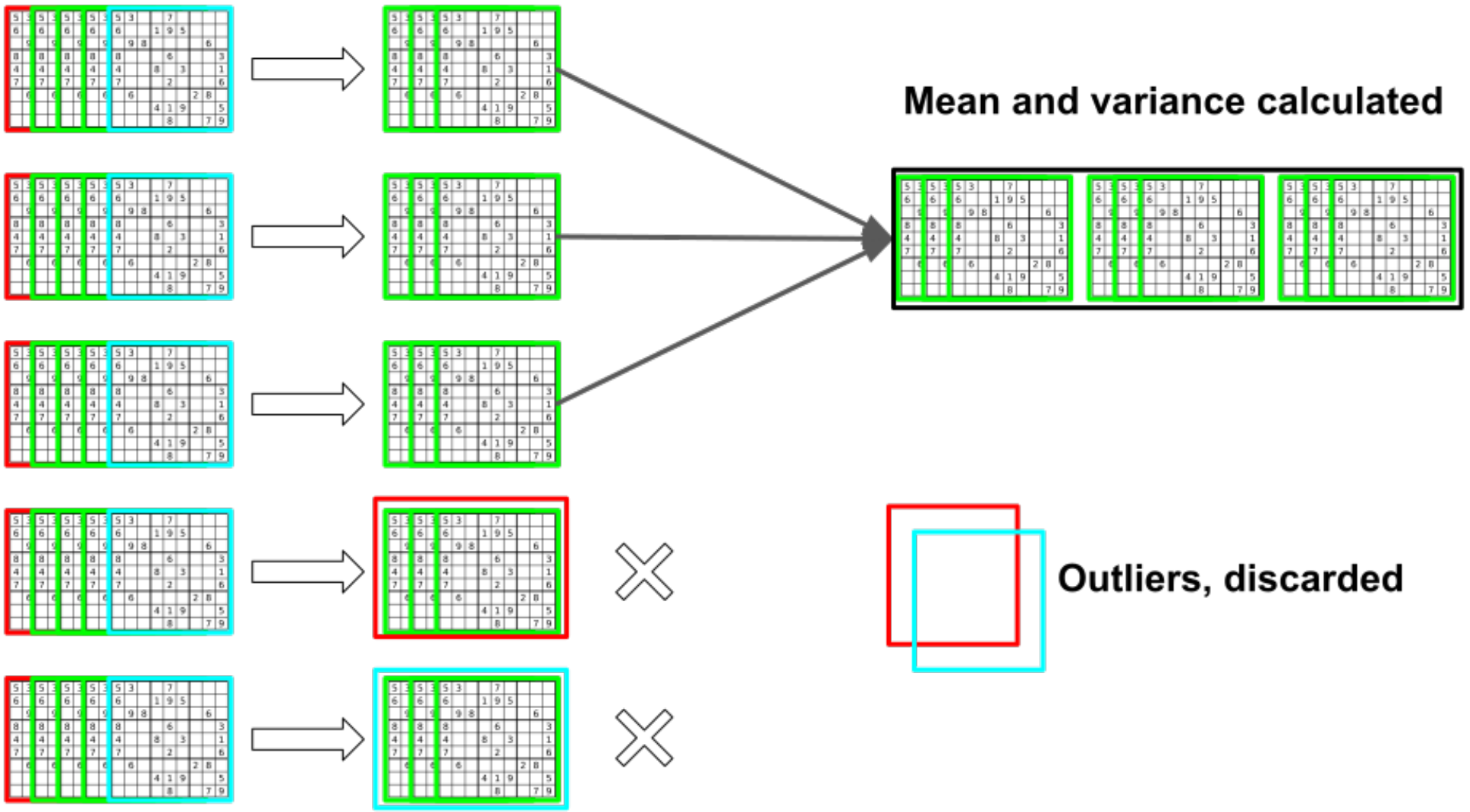


If the garbage collector is executed when running the SAT solver, the elapsed times increases by 12%!

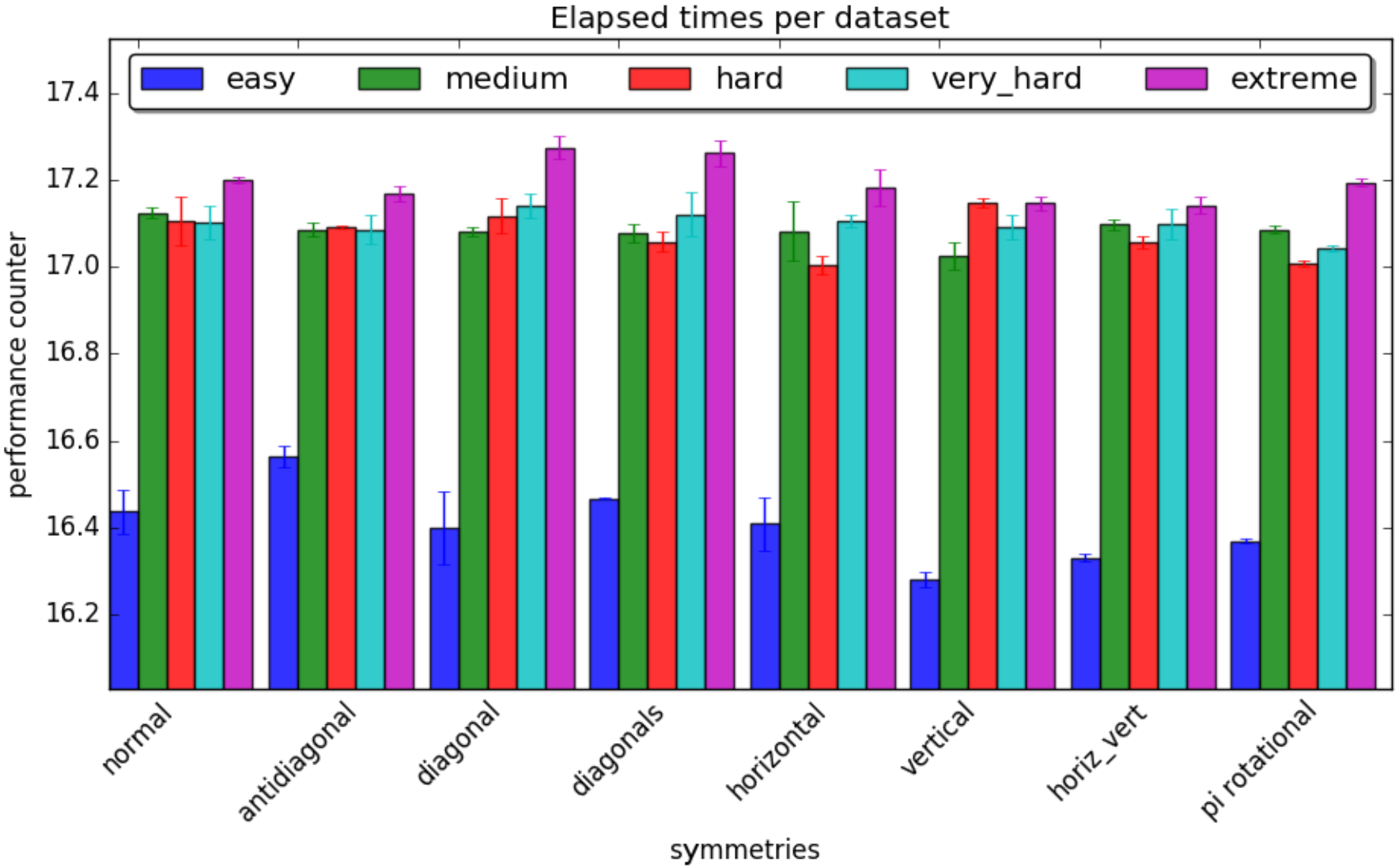
Experimental Procedure

- 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

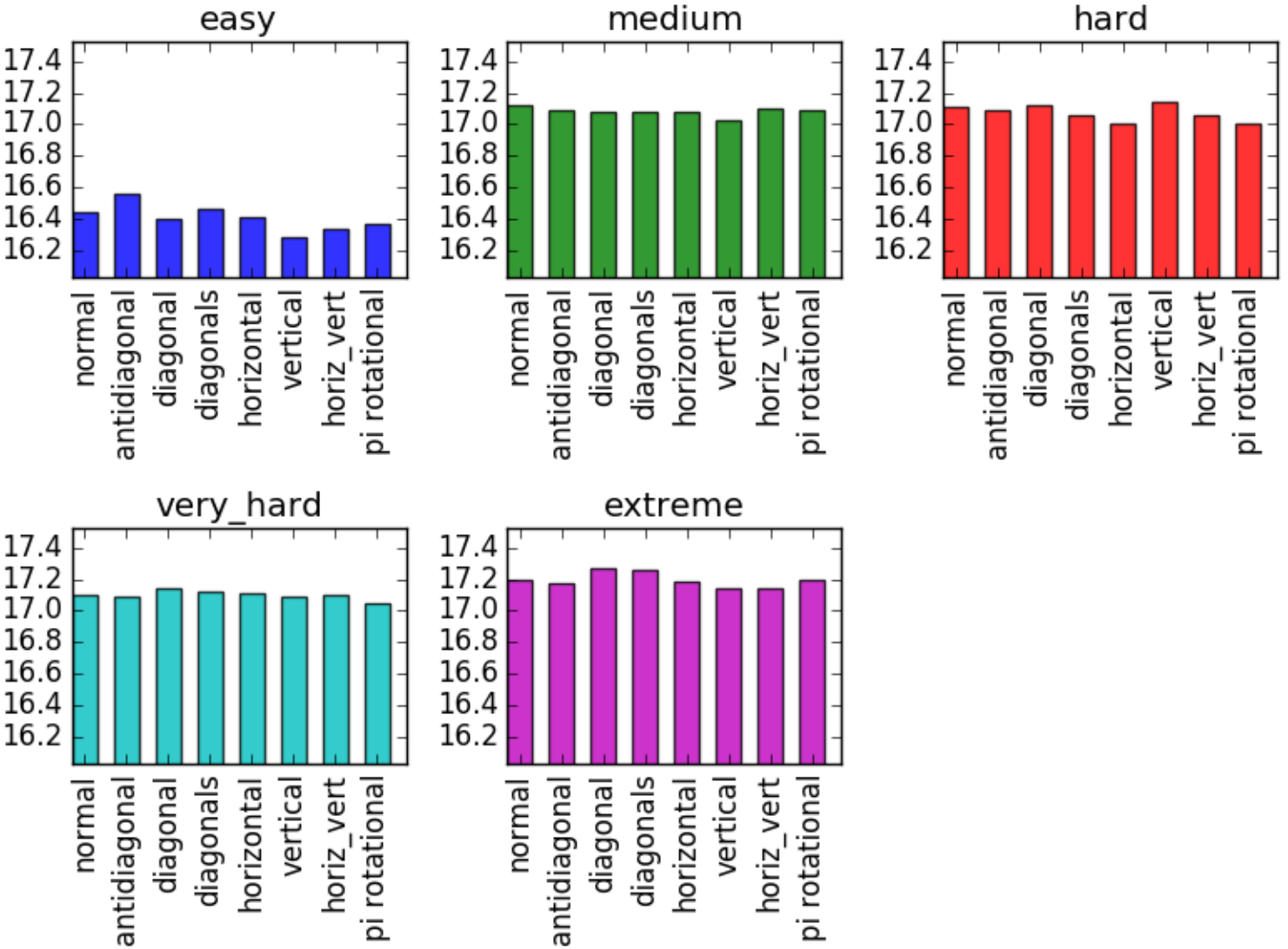
Experimental Procedure



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!